Cromemco Z80 Monitor

Cromemco™
incorporated
Tomorrow's Computers Today
280 BERNARDO AVE. MOUNTAIN VIEW, CA 94043

Part No. 023-0021

February 1980

Table of Contents

Introduction

The Z80 Monitor makes it possible to control computers which use the CROMEMCO ZPU^{tm} from a terminal keyboard. It includes executive commands to examine and change memory, make a binary or an ASCII dump of memory, move and compare blocks of memory, output a byte of data to any port, read, write, and punch nulls on binary paper tapes, program 2708 and 2704 PROMs using the CROMEMCO BYTE-SAVER, and initialize and control both serial ports on the CROMEMCO TUART.

Transfer of control to a program in memory can be commanded from the keyboard with up to five breakpoints set and with the initial contents of the ZPU registers specified. When a breakpoint is encountered during execution, control is transferred back to the monitor and the contents of all 22 ZPU registers are stored. These register values can be examined and changed before execution of the program is resumed.

Entry Points

The Z80 Monitor has three entry points. A cold-start entry at E000 hex selects bank 0 on CROMEMCO memory boards and UART A on the CROMEMCO TUART. It initializes the baud rate of the UART to match that of the terminal being used. In addition, it saves the contents of the Z80 registers I, N (IFF), S (SP) , X (IX) , Y (IY) , A' , B' , C' , D' , E' , F' , and H' (HL') in the user-register area which is part of the system stack. (If the Z80 stack pointer is pointing to RAM, then all registers except A and P (PC) will be saved.) The contents of these registers are restored when the monitor is exited by means of the GO command.

The warm-start entry point at E008 hex is provided so that the monitor can be re-entered without affecting the memory banks or the UART. The same registers are saved as for the cold-start entry point.

The third entry point is used by the breakpoint facility. Entry here saves the contents of all registers. Memory banks and UART are unaffected.

System Stack

The monitor does not require the user to address a RAM board at a special place in memory for its stack and working storage area. (However, if the breakpoint facility is used, there must be either RAM at locations 30, 31, and 32 hex or PROM with the data C3, 45, E0 hex at those locations.) The monitor finds the highest page of RAM active in the machine and places its stack and temporary storage area there. At least 60H or 96 bytes of this page must be reserved for system use. If the multiple command facility is used, each additional command in a command line requires an additional 20 hex or 32 bytes stack room. (See Multiple Commands.)

Command Format

The Z80 Monitor is controlled by one and two-character commands from the terminal keyboard. The format is free-form with respect to spaces.

In the following, DM is the Display Memory command and S is the Swath operator (see below). The four examples are equivalent commands. They display the contents of 100 hex bytes of memory beginning with location 1000 hex. (' (CR) ' indicates a carriage return.)

```
DM1000 10FF (CR)
DM1000S100(CR)
 D  M  1000     10FF (CR)
 D  M  1000 S   100  (CR)
```

When entering an address as a operand, only the last four digits typed in are retained. For example, '321000' is read as '1000'. Therefore, if a wrong digit is entered, continue typing until the last four digits are correct.

Only the last two digits typed are retained when a two-digit number such as a data byte is entered.

Swath Operator

There are two ways to specify the address range of many commands. The first is to simply list the beginning and ending addresses (and, where appropriate, the destination address). For example, the first command below programs the range 0 through 13FF into PROMs starting at E400. The second command displays the contents of memory between addresses E400 and E402.

```
PO  13FF E400
DME400   E402
```

Another way to do the same thing is to use the Swath operator, S, to specify the width of the address range rather than state the ending address explicitly.

```
PO  S1400 E400
DM  E400S3
```

Multiple Commands: The After Operator

The After operator, ' < ', can be used to place more than one command on a command line. All of the commands on the command line are executed before the monitor returns with its prompt ' : ', for a new command.

With this feature, the monitor can write an area of memory onto paper tape preceded and followed by a sequence of nulls without any undesirable carriage-returns or prompts inserted by the monitor.

Example 1

Assume that the terminal being used is a teletypewriter with paper tape punch. In order to write the contents of 400 hex bytes starting at 100 hex with a leader of 95 hex nulls and a trailer of 80 nulls, type:

```
:N80 < :W100 S 400 < :N 95 (CR)
```

where the colons are prompts provided by the monitor. Turn on the paper tape punch after typing the carriage-return in order to avoid writing it onto the tape.

There are several points to be made about the use of the After operator:

(a) The order of execution of the commands is from right to left. Hence, the name 'After' and the shape '<'.

(b) The After operator is logically equivalent to a carriage-return. Anywhere a carriage-return can reasonably appear in a command, the After operator may be used instead. However, no commands in the line are executed until an actual carriage-return is typed.

(c) If any of the GO commands appears in a multiple-command line, it must be the last command executed, i.e., the first command typed.

(d) Each additional command on a line adds from 10 to 20 hex bytes to the system stack size.

Example 2

Assume that we are using a CROMEMCO TUART I/O card with a console connected to UART A and with a paper tape reader and punch connected to the input and output, respectively, of UART B. Assume that the baud rate of UART B has already been set to that of the reader and punch. (See Baud Rates pg. 3.) We can copy a paper tape by switching the current UART to B, reading the tape into a memory buffer, writing a leader, writing the buffer to the punch, and finally switching the current UART back to A, the console, by typing:

```
:UA < :W0S2000 < :N80 < :R0S2000 < :UB (CR)
```

In this case, we can leave the reader and punch on all the time. There is no question of a carriage-return from the command line being punched onto the paper tape since two different UARTs are involved.

Perhaps we forgot to write nulls as a trailer to the output tape. After the prompt, ' : ', again appears on the console, we can rectify this by typing:

```
:U A < :N 80 < : U B (CR)
```

where, again, all colons are provided by the monitor.

Example 3

Suppose we wish to make three copies of the same PROM. Assume that the source is in RAM at location 0 and that we want three identical copies in PROMs located at E400, E800, and EC00 hex. The following command line will accomplish this:

```
:P0S400 EC00 < :P0S400 E800 < :P0S400 E400(CR)
```

Example 4

Either of the following will initialize the baud rate of a terminal connected to UART B of the TUART:

```
:I  < :UB   (CR)
:UA < :I  < :UB   (CR)
```

After entering one of these commands on the console connected to UART A, push CARRIAGE-RETURN on the other terminal until the monitor prompt ' : ' appears.

Example 5

Assume that we would like to take a brief nap to refresh ourselves but have no alarm clock. Assume further that two beeps of the console bell spaced 2.1 seconds apart are sufficient to wake us and that the console can run at 300 baud. Since the Display Memory command takes 63 characters to display 10 hex or 16 bytes of memory, at 300 baud it takes 2.1 seconds or 0.035 minutes to display 10 hex bytes.

Number of Bytes (hex)	Time (minutes)
10	0.035
640	3.5
C80	7.0
1900	14.0
3200	28.0
6400	56.0
C800	112.0

First, we re-initialize the UART by typing the following:

```
:I (CR)
```

Set the console baud rate to 300 and push the CARRIAGE-RETURN until the monitor issues its prompt, ' : '.

To ring the bell, output 7 to port 1. For a nap of 14 minutes:

```
:O 7 1 < :DM0S10 < :O7 1 < :DM0S1900 (CR)
```

Errors and Escapes

When the monitor detects an error condition, the command is aborted, all breakpoints are cleared, and a '?' is printed followed by the prompt ' : ' for the next command.

Any command may be aborted from the keyboard either when the monitor is requesting further input, or during print-out, by depressing either the ESCAPE or the ALT MODE key. CONTROL-SEMI-COLON, CONTROL-SHIFT-'K', and '}' may also work.

Input and Output

The monitor assumes that a data transfer occurs on I/O port 1. Status flags are transmitted over input port 0. The data-available flag is on bit 6 of input port 0. The transmitter-buffer-empty flag is on bit 7 of input port 0. Both flags are active high.

To use the CROMEMCO TUART with the monitor, set switches 1, 7, and 9 of the 10-position TUART switch OFF, all others ON. The currently selected UART uses I/O port 1 for date transfer and input port 0 for status flags. The UART which is not current uses I/O port 51 hex for date transfer and input port 50 hex for status flags. (The UARTs are selected by means of the UART command.)

The following locations may be changed for different I/O conventions:

Status port number (00): E00F, E020
Input data port number (01): E014
Output data port number (01): E027
Input-data-available mask (40): E011
Output-transmitter-buffer-empty mask (80): E022

For active-low status flags change locations E019 and E379 from 28 hex to 20 hex and change location E120 from 20 hex to 28 hex.

Baud Rates and UART Selection

When the monitor is entered at E000 hex, the cold-start entry point, push CARRIAGE-RETURN until the monitor responds with:

CROMEMCO ZM1.4

The monitor is capable of selecting 19200, 9600, 4800, 2400, 1200, 300, 150, or 110 baud when used with the CROMEMCO TUART I/O board.

The maximum number of carriage-returns required to select any of these baud rates is four. (Two carriage-returns are required for any UART with a fixed baud rate.)

The baud rate can also be changed by using the Initialize command (see page 5).

Some peripheral devices such as paper tape readers or punches may have no keyboards. The TUART baud rate can also be set by outputting a data byte from the following table to port 0 for the currently selected UART or to port 50 hex for the unselected UART. (To make UART B current, output 80 hex to port 4. For UART A, output 0 to port 54 hex. UART selection can also be accomplished by means of the monitor's UART command, U).

Baud Rate	Data Byte
110	01
150	82
300	84
1200	88
2400	90
4800	A0
9600	C0

The baud rate can be octupled by outputting 10 hex to port 2 for the selected UART or to port 52 hex for the other UART. Outputting 0 to these ports brings the baud rate back to normal.

Interrupts

The monitor can be used to enable interrupts in the Z80. This is done by changing the value of the N register to 1 by using the Substitute Register command, SN. (The N register stores the value of the Z80 interrupt flip-flop at the time the monitor is entered.) Then interrupts will be enabled when one of the Go commands is given.

Note, however, that the interrupt mask registers on the TUART must have been set previously, either by a user program or by the monitor. (If this is not done, then an immediate interrupt will be generated because the print buffer is empty.) To mask out all interrupts output 0 to port 3 for the current UART and to port 53 hex for the other UART.

The mask bit corresponding to each of the possible interrupts is given in the following table:

Bit	Interrupting Device
0	Timer 1
1	Timer 2
2	Sens (external)
3	Timer 3
4	Receiver Data Available
5	Transmitter Buffer Empty
6	Timer 4
7	Timer 5 or external

For example, to allow only interrupts from the serial input port and from Timer 1 on the current UART, output 11 hex to port 3 and 0 to port 53 hex.

Installing the Monitor

The Cromemco Z80 Monitor is supplied in a 2708 ROM. This ROM may be installed on any Cromemco PROM memory board and must be addressed at E000 hex.

Using the Monitor

Set the power-on jump switch on the Cromemco ZPU card to E (1110 binary). Whenever the computer is reset, control will then immediately pass to the monitor.

If the ZPU is used with the Cromemco TUART I/O card, depress CARRIAGE-RETURN two to four times. This will set the UART on the serial interface card to the baud rate of the terminal being used.

When used with a serial interface card with baud rate fixed to that of the terminal, simply depress CARRIAGE-RETURN twice. The monitor will then respond:

```
CROMEMCO ZM1.4
```

followed by a prompt ' : '. The monitor is then ready to accept commands from the keyboard.

COMMANDS

DISPLAY MEMORY

[1] DM beginning-addr ending-addr (CR)

or

DM beginning-addr S swath-width (CR)

The contents of memory are displayed in hexadecimal form. Each line of the display is preceded by the address of its first byte. Example:

```
:DM100 S3
0100: C3 34 7F
```

DISPLAY REGISTERS

[2] DR (CR)

When the monitor is re-entered from a breakpoint, the contents of all the Z80 registers are stored in an area called the user-register area. (When the monitor is entered via reset or the warm-start entry point, all registers except A, B, C, D, E, F, HL, and P are saved in the user-register area. However, if the stack pointer is pointing to RAM, then all but A and P will be saved.)

DR causes these stored registers to be displayed in the following format:

```
A=01 B=12 C=34 D=56 E=78 F=9A HL=BCDE
I=F0 N=00 P=1234 S=5678 X=9ABC Y=DEF0
A'23 B'45 C'67 D'89 E'AB F'CD HL'EF01
```

If interrupts were enabled when the monitor was entered, then N=1. Otherwise, N=0.

The flag registers, F and F', are packed as follows:

```
S,Z,x,H x,P/V,N,C
```

i.e., sign, zero, (unknown), half-carry, (unknown), parity or overflow, subtraction, and carry flags.

GO

[3] G (CR)

The Z80 registers are loaded with the values saved in the user-register area. (These are the values displayed with the DR command.) Execution then resumes at the location contained in the user-program-counter, P.

[4] G starting-addr (CR)

This command is exactly like [3] except that the user-program-counter, P, is first loaded with starting-address. Thus, execution begins at starting-address.

GO WITH BREAKPOINTS SET

[5] G / breakpoint-addr-1 breakpoint-addr-2 . . . (CR)

[6] G starting-addr / brkpt-addr-1 brkpt-addr-2 . . . (CR)

Commands [5] and]6] are like [3] and [4], respectively, except that breakpoints are set at breakpoint-address-1, breakpoint-address-2, etc.

When a breakpoint is encountered in the execution of the user program, the monitor is re-entered. All registers are saved in the user register area (which is part of the system stack), the address of the breakpoint is printed, and all breakpoints are cleared (i.e., the user program is restored to its original state). Finally, the prompt, ' : ' is issued for the next command from the keyboard. Note the following about the use of breakpoints:

(a) Breakpoints can only be set in programs residing in RAM. This is because the monitor inserts a RST 48 instruction (F7 hex) at each breakpoint location. (The original contents of these locations are saved so that they can later be restored.)

(b) Up to five breakpoints can be set. If an attempt is made to set a sixth breakpoint, the monitor will print a question mark to indicate error, erase all breakpoints, and prompt for a new command.

(c) When a breakpoint is set, the monitor inserts a 3-byte jump instruction at location 30 hex. This means that locations 30, 31, and 32 hex are not available to the user program when breakpoints are used.

(d) The monitor temporarily uses ten bytes on the user's stack in executing a breakpoint. The area reserved for the user's stack must, therefore, be at least ten bytes larger than that required for the user's program.

(e) If breakpoints are set in a program and the computer is reset and the monitor re-entered before any breakpoint is reached in the execution of the program, then the breakpoints will have to be removed from the program by means of the Substitute Memory command, SM. However, if any breakpoint is reached, all breakpoints are automatically cleared by the monitor.

INITIALIZE BAUD RATE

[7] I (CR)

After the CARRIAGE-RETURN is typed, change the baud rate of the terminal to the desired value and then push the CARRIAGE-RETURN until the monitor responds with its prompt, ' : '.

The monitor is capable of selecting 19200, 9600, 4800, 2400, 1200, 300, 150, or 110 baud when used with the Cromemco TUART I/O board. The maximum number of carriage-returns required to select any of these baud rates is four.

The command is particularly useful for setting the baud rate of the second serial port on the TUART. (See Multiple Commands.)

MOVE

[8] M source-addr source-end destination-addr (CR)

or

M source-addr S swath-width destination-addr (CR)

Move the contents of memory beginning with source-address and ending with source-end to destination-address. After the move, the monitor verifies that source and destination are the same. This will result in a print-out of discrepancies which are not really errors after certain types of overlapping moves. However, this print-out can be terminated by depressing ESCAPE or ALT MODE.

The Move command can be used to fill a block of memory with a constant. For example, to enter zeros between locations 100 and 108, use the Substitute Memory command to enter 0 at location 100, and then move 100 through 107 to 101:

```
M100  107  101
or
M 100  S  8   101
```

Care should be taken not to overwrite the system stack which resides in the top of active RAM. (See System Stack.)

NULLS

[9] N hex-number (CR)

Write hex-number nulls to the current device. This command is used to punch leaders and trailers on paper tape. (See Multiple Commands.)

OUTPUT

[10] O data-byte port-number (CR)

Outputs data to a port. One use of this command is to select banks on Cromemco memory boards. When the monitor is first entered on power-up or reset, it selects bank 0 and turns off all other memory banks.

Either a software output or a monitor output to port 40 hex serves to change the bank selection. To select bank n, output a byte with bit n high. To select two banks, n and m, output a byte with both bits n and m high.

Bank	Output byte
0	01
1	02
2	04
3	08
4	10
5	20
6	40
7	80

For example, the first command selects bank 5 and the second selects banks 4 and 5.

```
O 20 40
O 30 40
```

PROGRAM

[11] P source-addr source-end destination-addr (CR)
or
P source-addr S swath-width destination-addr (CR)

Program from source-address through source-end into PROMS beginning at destination-address.

If the length of the source is not a multiple of 400H (1024 decimal) or if the destination does not begin at 400H boundary, the monitor will reject the command. (Multiples of 400H end in '000', '400', '800', or 'C00'.)

Any number of 2708 or 2704 PROMS can be programmed in the execution of one command as long as there are enough BYTESAVERS to contain them. Each PROM is verified with its source after all are programmed and any discrepancies are printed out. If there are none, the prompt ' : ' is issued and the monitor awaits the next command.

Software can be loaded into a PROM in as small increments as you desire provided it is added to previously unused areas of the PROM.

This is done by first using the Move command, M, to transfer the current contents of the PROM down to RAM, adding the new software to an area of RAM which corresponds to the unused portion of the PROM and finally using the Program command, P, to re-program the PROM with the result.

Although the entire PROM must always be programmed, it never hurts to re-write the same data over again.

In general, a 1 may be written over a 1, a 0 over either a 1 or a 0, but the only way to change 0's to 1's is to erase the PROM with appropriate UV light. (See the BYTESAVER manual for details.)

READ

[12] R destination-addr destination-end (CR)
or
R destination-addr S swath-width (CR)

Read binary or ASCII input from paper tape reader or console and store in memory from destination-address through destination-end. After destination-end has been filled, the monitor prompts for the next command.

SUBSTITUTE MEMORY

[13] SM address (CR)

Substitute Memory desplays the contents of address and outputs a dot, ' . ', as a prompt for the substituted value. If no change is desired, type a space or another dot. Otherwise, enter the new value. The monitor accepts hex digits until it gets a delimiter, such as a space, dot, or carriage-return retaining the last two digits entered as the value. Unless the delimiter is a carriage-return, the monitor outputs the contents of the next sequential memory location with a dot prompt. A carriage-return terminates the command.

SUBSTITUTE REGISTER

[14] S register-name (CR)

Register-name may be A, B, C, D, E, F, H (HL), I, N (state of the Z80 interrupt flip-flop), P (PC), S (SP), A', B', C', D', E', F', H' (HL'), X (IX), or Y (IY).

This command prints the name of the user-register requested, displays its contents, outputs a dot, ' . ', as a prompt for the substituted value. If no change is desired, type a space or another dot. Otherwise, enter the new value. The monitor accepts hex digits until it gets a delimiter such as space, dot, or carriage-return retaining the last two digits (four digits for a 2-byte register). Unless the delimiter is a carriage-return, the monitor prints the name and contents of the next register followed by the dot prompt. A carriage return terminates the command.

UART SELECT

[15] U device-name (CR)

Device-name may be A or B. The Cromemco TUART has two UARTs. When the monitor is entered via reset, UART A is selected for its input/output channel. This command allows the user to change the UART selection. It is often used in the multiple command mode (see page 2).

VERIFY

[16] V source-addr source-end destination-addr (CR)
or
V source-addr S swath-width destination-addr (CR)

Verify that the block of memory between source-address and source-end contains the same values as the block beginning at destination-address. The addresses and contents are printed for each discrepancy found (unless the print-out is terminated by ESCAPE or ALT MODE).

This command works by reading bytes from the source and destination and comparing them. If a discrepancy is found, the memory is read again for print-out. Thus, it can happen that a discrepancy is printed-out with the source and destination contents indicated to be the same. This is caused by a defective memory element.

WRITE

[17] W source-addr source-end (CR)

or

W source-addr S swath-width (CR)

Write binary or ASCII output from source-address through source-end to the current device (selected by the UART command). After source-end has been written, the monitor prompts for the next command.

The Write command is useful for punching binary or ASCII paper tapes of the contents of memory and for looking at the ASCII contents of memory on the console.

When punching a paper tape, it is usually desirable to punch series of nulls as leader and trailer. This can best be done in conjunction with the Null command and the After operator. (See Multiple Commands for examples of this usage.)

Program Listing

```
                    ;
                    ;
      (ØØØØ)        STAT:    EQU     Ø              ;STATUS PORT, DEVICE A
      (ØØØ1)        DATA:    EQU     1              ;DATA PORT, DEVICE A
      (ØØØ2)        ACMNDP:  EQU     2              ;COMMAND PORT, DEV. A
      (ØØØØ)        ABAUDP:  EQU     Ø              ;BAUD PORT, DEVICE A
      (ØØØ4)        APARLP:  EQU     4              ;PARALLEL PORT, DEV. A
      (ØØ52)        BCMNDP:  EQU     52H            ;COMMAND PORT, DEV. B
      (ØØ54)        BPARLP:  EQU     54H            ;PARALLEL PORT, DEV. B
      (ØØ4Ø)        DAV:     EQU     4ØH            ;DATA-AVAILABLE MASK
      (ØØ8Ø)        TBE:     EQU     8ØH            ;XMITTER-BUF-EMPTY MSK
                    ;
      (ØØØ5)        NBRKPT:  EQU     5              ;ALLOW ROOM FOR
      (ØØ16)        BPSTOR:  EQU     NBRKPT*4+2     ;BREAKPOINT STORAGE
      (ØØ16)        TEMPS:   EQU     BPSTOR
      (ØØØB)        BPMRK:   EQU     ØBH            ;USED TO MARK THE SET-
                    ;                               ;TING OF A BP IN BPSTOR.
      (ØØ3Ø)        RSTLC:   EQU     3ØH            ;RST LOCATION
      (ØØØØ)        CASE:    EQU     Ø              ;(REQUIRES UPPER-CASE)
      (ØØØ5)        B2F:     EQU     5              ;2-BYTE FLAG
      (ØØØ6)        PF:      EQU     6              ;PRIME-ABLE REG FLAG
      (ØØØ7)        CRF:     EQU     7              ;CRLF FLAG
                    ;
      (ØØØD)        CR:      EQU     ØDH
      (ØØØA)        LF:      EQU     ØAH
      (ØØ1B)        ESC:     EQU     1BH
      (ØØ7D)        ALT:     EQU     7DH
                    ;
                    ; DISPLACEMENTS FROM IX OF HI BYTE OF REG PAIRS
                    ;
                    ;
      (FFFF)        DUPC:    EQU     -1
      (FFFD)        DUAF:    EQU     -3
      (FFFB)        DUBC:    EQU     -5
      (FFF9)        DUDE:    EQU     -7
      (FFF7)        DUHL:    EQU     -9
      (FFF5)        DUSP:    EQU     -11
      (FFF3)        DUIX:    EQU     -13
      (FFF1)        DUIY:    EQU     -15
      (FFEF)        DUIN:    EQU     -17            ;I & THE INTERRUPT FF
      (FFED)        DUAF2:   EQU     -19
      (FFEB)        DUBC2:   EQU     -21
      (FFE9)        DUDE2:   EQU     -23
      (FFE7)        DUHL2:   EQU     -25
                    ;
      (ØØ1A)        LENRGS:  EQU     DUPC-DUHL2+2
                    ;
                    ;
                    ;
                    ;
EØØØ                         ORG     ØEØØØH
                    ;
                    ; ENTER THE MONITOR FROM RESET.
                    ; COLD START ENTRY. INITIALIZES THE UART
                    ; AND ZEROES THE BREAKPOINT STACK POINTER.
                    ; ALTERS THE A-REGISTER.  SAVES ALL OTHER
                    ; REGISTERS EXCEPT THE PROGRAM COUNTER,
                ; BUT DOES NOT DISPLAY THEM.
                ;
E000   3E01     CSTART: LD      A,1
E002   D340             OUT     40H,A           ;SELECT BANK 0
E004   F5               PUSH    AF              ;SIMULATE UPC
E005   F5               PUSH    AF              ;USER-F-REGISTER
E006   1842             JR      COMMON
                ;
                ;
                ;
                ; WARM START ENTRY.  INITIALIZES THE BREAKPOINT
                ; STORAGE POINTER.  SAVES ALL REGISTERS EXCEPT
                ; THE PROGRAM COUNTER, BUT DOES NOT DISPLAY THEM.
                ;
E008   F5       WSTART: PUSH    AF              ;SIMULATE UPC
E009   F5               PUSH    AF              ;UAF
E00A   3E80             LD      A,80H           ;FLAG:
E00C   183C             JR      COMMON          ;WARM-START ENTRY
                ;
                ;
                ; CHECK INPUT & RETURN WITH DATA IF READY.
                ;
E00E   DB00     CHKIN:  IN      A,STAT
E010   E640             AND     DAV
E012   C8               RET     Z
E013   DB01             IN      A,DATA
E015   C9               RET
                ;
                ;
                ; GET CHARACTER FROM INPUT.
                ;
E016   CD0EE0   GBYTE:  CALL    CHKIN
E019   28FB             JR      Z,GBYTE
E01B   E67F             AND     7FH
E01D   C9               RET
                ;
                ;
                ; PRINT CHARACTER.
                ;
E01E   F5       PBYTE:  PUSH    AF
E01F   DB00     PBY1:   IN      A,STAT
E021   E680             AND     TBE
E023   28FA             JR      Z,PBY1
E025   F1               POP     AF
E026   D301             OUT     DATA,A
E028   C9               RET
                ;
                ;
                ; SELECT DEVICE A & INITIALIZE ITS BAUD RATE.
                ; ENTER WITH A=1.
                ;
E029   D354     INIT:   OUT     BPARLP,A        ;SELECT DEVICE A
E02B   D352             OUT     BCMNDP,A        ;RESET DEVICE B
                ;                               ;[CONTINUE BELOW]
                ;
                ;
                ; INITIALIZE BAUD RATE OF THE CURRENT DEVICE.
```

```
                        ;
                        ; PUSH CARRIAGE-RETURN TO SELECT THE PROPER BAUD
                        ; RATE FOR THE CURRENT TERMINAL.  (THE MAXIMUM
                        ; NUMBER OF CARRIAGE-RETURNS REQUIRED IS FOUR.)
                        ;
                        ; WITHE THE CROMEMCO TUART ANY OF THE FOLLOWING
                        ; BAUD RATES CAN BE SELECTED:
                        ; 192ØØ, 96ØØ, 48ØØ, 24ØØ, 12ØØ, 3ØØ, 15Ø, 11Ø.
                        ;
                        ; WITH THE 3P+S:  24ØØ, 3ØØ, 11Ø.
                        ;
                        ; TWO CARRIAGE-RETURNS ARE REQUIRED FOR
                        ; ANY UART WITH A FIXED BAUD RATE.
                        ;
EØ2D    21A3E3          INITBAUD: LD      HL,BAUDRS
EØ3Ø    ØEØØ                      LD      C,ABAUDP
EØ32    3E11                      LD      A,11H          ;OCTUPLE THE CLOCK
EØ34    D3Ø2            IT1:      OUT     ACMNDP,A       ;& RESET CURRENT DEVICE
EØ36    EDA3                      OUTI
EØ38    CD16EØ                    CALL    GBYTE
EØ3B    CD16EØ                    CALL    GBYTE
EØ3E    FEØD                      CP      CR
EØ4Ø    3EØ1                      LD      A,1            ;SLOW THE CLOCK
EØ42    2ØFØ                      JR      NZ,IT1
EØ44    C9                        RET
                        ;
                        ;
                        ; BREAKPOINT ENTRY. INITIALIZES NOTHING.
                        ; SAVES ALL REGISTERS AND DISPLAYS THEM.
                        ;
EØ45    E3              SVMS:     EX      (SP),HL        ;ADJUST BRKPT
EØ46    2B                        DEC     HL             ;RET ADDR
EØ47    E3                        EX      (SP),HL
EØ48    F5                        PUSH    AF             ;UAF
EØ49    97                        SUB     A              ;FLAG:
                        ;                                ;BREAKPOINT ENTRY;
                        ;
                        ;
EØ4A    C5              COMMON:   PUSH    BC             ;UBC
EØ4B    47                        LD      B,A            ;ENTRY FLAG
EØ4C    D5                        PUSH    DE             ;UDE
EØ4D    E5                        PUSH    HL             ;UHL
                        ;
                        ; PLACE SYS STACK AT HIGHEST PAGE OF
                        ; AVAILABLE RAM.
                        ; ALLOW ROOM FOR TEMP STORAGE.
                        ;
EØ4E    21E9ØØ                    LD      HL,ØØFFH-TEMPS
EØ51    25              COM1:     DEC     H
EØ52    7E                        LD      A,(HL)
EØ53    34                        INC     (HL)
EØ54    BE                        CP      (HL)           ;DID IT CHANGE?
EØ55    28FA                      JR      Z,COM1
EØ57    35                        DEC     (HL)           ;YES. RESTORE IT.
                        ;
EØ58    78                        LD      A,B            ;ENTRY FLAG
EØ59    EB                        EX      DE,HL
```

```
E05A   21Ø9ØØ              LD      HL,9
E05D   39                  ADD     HL,SP             ; -> UPC, HI BYTE
E05E   Ø1ØAØØ              LD      BC,1Ø
E061   EDB8                LDDR
                   ;
E063   13                  INC     DE                ;-> UHL,LO ON SYS STK
E064   EB                  EX      DE,HL
E065   F9                  LD      SP,HL             ;CURRENT SYS SP
E066   EB                  EX      DE,HL
E067   Ø1ØBØØ              LD      BC,DUPC-DUHL+3
E06A   Ø9                  ADD     HL,BC             ;HL = USER SP
E06B   E5                  PUSH    HL                ;USP
E06C   DDE5                PUSH    IX                ;UIX
E06E   FDE5                PUSH    IY                ;UIY
E07Ø   EB                  EX      DE,HL
E071   Ø9                  ADD     HL,BC
E072   4D                  LD      C,L               ;SAVE
E073   2B                  DEC     HL
E074   E5                  PUSH    HL
E075   DDE1                POP     IX
E077   FEØ1                CP      1                 ;ENTRY?
E079   38Ø7                JR      C,COM3            ;SKIP IF VIA BP.
E07B   71                  LD      (HL),C            ;BP PNTR, LO BYTE
E07C   23                  INC     HL
E07D   36ØØ                LD      (HL),Ø            ;BP-STACK ENDMARK
                   ; INITIALIZE THE TUART IF ENTRY WAS VIA RESET.
                   ; (A CONTAINS 1.)
                   ;
E07F   CC29EØ              CALL    Z,INIT
                   ;
E082   ED57        COM3:   LD      A,I
E084   67                  LD      H,A
E085   2EØØ                LD      L,Ø
E087   E28BEØ              JP      PO,COM4
E08A   2C                  INC     L
E08B   E5          COM4:   PUSH    HL                ;UIN
E08C   Ø8                  EX      AF,AF'
E08D   F5                  PUSH    AF                ;UAF'
E08E   Ø8                  EX      AF,AF'
E08F   D9                  EXX
E09Ø   C5                  PUSH    BC                ;UBC'
E091   D5                  PUSH    DE                ;UDE'
E092   E5                  PUSH    HL                ;UHL'
E093   D9                  EXX
                   ;
                   ; IF CY IS SET, ENTRY WAS VIA A BREAKPOINT
E094   21FØE3              LD      HL,HEAD
E097   D4ØFE2              CALL    NC,PMSG
E09A   Ø1865Ø              LD      BC,[['P'+CASE] SHL 8]+86H ;IF BP ENTRY,
E09D   DC23E3              CALL    C,SUBR3           ;DISPLAY THE PC.
                   ;
                   ;
                   ;CLEAR ALL BREAKPOINTS
                   ;
                   ;
E0AØ   DDE5        CLBP:   PUSH    IX
E0A2   E1                  POP     HL                ;POINTS TO BPSP,LO
```

```
EØA3   6E          Ø23Ø          LD       L,(HL)          ;BPSP NOW IN HL
                   Ø231 ;
EØA4   7E          Ø232 CL1:     LD       A,(HL)          ;BP STK EMPTY?
EØA5   FEØB        Ø233          CP       BPMRK           ;IF BPMRK, BP IS SET
EØA7   2ØØA        Ø234          JR       NZ,CL2
                   Ø235 ;
EØA9   34          Ø236          INC      (HL)            ;BP-ERASED MARK
EØAA   2B          Ø237          DEC      HL
EØAB   56          Ø238          LD       D,(HL)
EØAC   2B          Ø239          DEC      HL
EØAD   5E          Ø24Ø          LD       E,(HL)
EØAE   2B          Ø241          DEC      HL
EØAF   EDA8        Ø242          LDD                      ;RESTORE MEM CONTENTS
EØB1   18F1        Ø243          JR       CL1
                   Ø244 ;
EØB3   7D          Ø245 CL2:     LD       A,L
EØB4   2B          Ø246          DEC      HL
EØB5   77          Ø247          LD       (HL),A          ;ADJUST BPSP
                   Ø248 ;
EØB6   11E6FF      Ø249          LD       DE,-LENRGS      ;FOR THE BENEFIT
EØB9   19          Ø25Ø          ADD      HL,DE           ;OF ERROR & ESCPE
EØBA   F9          Ø251          LD       SP,HL           ;RE-INITIALIZE SP
                   Ø252 ;
                   Ø253 ;
                   Ø254 ; GET 1-BYTE COMMAND.
                   Ø255 ; RETURNS VALUE IN HL & JUMPS TO THAT ADDR.
                   Ø256 ;
EØBB   CD4DE1      Ø257          CALL     CRLF
EØBE   11BEEØ      Ø258 CMND:    LD       DE,CMND         ;SET-UP RETURN
EØC1   D5          Ø259          PUSH     DE
EØC2   21AEE3      Ø26Ø CMND1:   LD       HL,PRMPT        ;RE-ENTRY POINT
EØC5   CDØFE2      Ø261          CALL     PMSG            ;FOR RECURSION
                   Ø262 ; HL NOW PNTS TO THE COMMAND TABLE.
                   Ø263 ;
                   Ø264 ; GET THE COMMAND.
                   Ø265 ; DE GETS THE FIRST ALPHA CHAR LESS 'D'.
                   Ø266 ;
EØC8   CDDDE1      Ø267          CALL     SKSGØ           ;GET NON-SPACE
EØCB   C8          Ø268          RET      Z               ;IF CR, IGNORE.
EØCC   D644        Ø269          SUB      'D'+CASE        ; < 'D'?
EØCE   3815        Ø27Ø          JR       C,ERROR
EØDØ   FE14        Ø271          CP       'W'-'D'+1       ; > 'W'?
EØD2   3Ø11        Ø272          JR       NC,ERROR
EØD4   5F          Ø273          LD       E,A
EØD5   16ØØ        Ø274          LD       D,Ø
                   Ø275 ;
EØD7   4A          Ø276          LD       C,D             ;INITIALIZE FOR SUBR
EØD8   EB          Ø277          EX       DE,HL
EØD9   29          Ø278          ADD      HL,HL           ;TIMES 2
EØDA   19          Ø279          ADD      HL,DE           ; + TBL ADDR
EØDB   5E          Ø28Ø          LD       E,(HL)
EØDC   23          Ø281          INC      HL
EØDD   56          Ø282          LD       D,(HL)
EØDE   EB          Ø283          EX       DE,HL
EØDF   CDDDE1      Ø284          CALL     SKSGØ           ;NEXT CMND GHAR
EØE2   FE4D        Ø285          CP       'M'+CASE        ;(USED IN SUBST & DISPL)
EØE4   E9          Ø286          JP       (HL)
```

```
                 Ø287 ;
                 Ø288 ;
                 Ø289 ; ERROR & ESCAPE. RETURNS TO CMND WITH SP
                 Ø290 ; POINTING TO SAVED-REG AREA (UHL').
                 Ø291 ;
EØE5  3E3F       Ø292 ERROR:  LD     A,'?'
EØE7  CD12E1     Ø293         CALL   PCHR
EØEA  18B4       Ø294 ESCPE:  JR     CLBP           ;CLEAR ANY BRKPTS
                 Ø295 ;
                 Ø296 ;
                 Ø297 ; PROGRAM PROMS. ABORTS IF DESTINATION
                 Ø298 ; IS NOT ON A 1K (4ØØH) BOUNDARY, OR IF SWATH
                 Ø299 ; WIDTH IS NOT A MULTIPLE OF 1K.
                 Ø3ØØ ;
                 Ø3Ø1 ;
EØEC  CDA5E1     Ø3Ø2 PROG:   CALL   L3NCR
EØEF  78         Ø3Ø3         LD     A,B            ;ARE INCREMENT &
EØFØ  B2         Ø3Ø4         OR     D              ;DESTINATION BOTH
EØF1  E6Ø3       Ø3Ø5         AND    3              ;MULTIPLES OF
EØF3  B1         Ø3Ø6         OR     C              ;1Ø24?
EØF4  B3         Ø3Ø7         OR     E
EØF5  2ØEE       Ø3Ø8 ERRV1:  JR     NZ,ERROR       ;ERROR VECTOR
                 Ø3Ø9 ;
EØF7  E5         Ø31Ø         PUSH   HL             ;SOURCE
EØF8  214ØØ1     Ø311         LD     HL,32Ø         ;# OF ITERATIONS
EØFB  E3         Ø312 PR1:    EX     (SP),HL
EØFC  CD1AE2     Ø313         CALL   MVE            ;MOVE IT
EØFF  E3         Ø314         EX     (SP),HL
E1ØØ  2B         Ø315         DEC    HL             ;ITERATION CT
E1Ø1  7C         Ø316         LD     A,H
E1Ø2  B5         Ø317         OR     L
E1Ø3  2ØF6       Ø318         JR     NZ,PR1
E1Ø5  E1         Ø319         POP    HL
E1Ø6  1861       Ø32Ø         JR     VRFY           ;VERIFY IT
                 Ø321 ;
                 Ø322 ;
                 Ø323 ; PRINT THE 2 BYTES IN (HL) & (HL-1).
                 Ø324 ; DECREMENTS HL BY 2.  ALTERS A.
                 Ø325 ; PRESERVES OTHER REGS.
                 Ø326 ;
E1Ø8  CDECE1     Ø327 P2NMS:  CALL   PNM
E1ØB  2B         Ø328         DEC    HL
E1ØC  CDECE1     Ø329         CALL   PNM
E1ØF  2B         Ø33Ø         DEC    HL             ;(CONTINUE BELOW)
                 Ø331 ;
                 Ø332 ;
                 Ø333 ; PRINT SPACE.  ALTERS A.
                 Ø334 ;
E11Ø  3E2Ø       Ø335 SPACE:  LD     A,2ØH          ;(CONTINUE BELOW)
                 Ø336 ;
                 Ø337 ;
                 Ø338 ; PRINT THE CHARACTER IN THE A-REGISTER.
                 Ø339 ; (CHKS INPUT FOR ESC.) PRESERVES ALL REGS.
                 Ø34Ø ;
E112  F5         Ø341 PCHR:   PUSH   AF             ;SAVE THE CHAR
E113  E67F       Ø342 PC1:    AND    7FH
E115  FE1B       Ø343         CP     ESC
```

```
E117  28D1       Ø344          JR      Z,ESCPE
E119  FE7D       Ø345          CP      ALT              ;ALT MODE?
E11B  28CD       Ø346          JR      Z,ESCPE
E11D  CDØEEØ     Ø347          CALL    CHKIN
E12Ø  2ØF1       Ø348          JR      NZ,PC1
                 Ø349 ;
E122  F1         Ø35Ø PC2:     POP     AF
E123  E5         Ø351          PUSH    HL
E124  F5         Ø352          PUSH    AF
E125  E67F       Ø353          AND     7FH
E127  CD1EEØ     Ø354          CALL    PBYTE
E12A  21ABE3     Ø355          LD      HL,LFNN
E12D  FEØD       Ø356          CP      CR
E12F  CCØFE2     Ø357          CALL    Z,PMSG
E132  FE3C       Ø358          CP      '<'              ;RECURSIVE CALL
E134  2ØØB       Ø359          JR      NZ,PC3           ;ON CMND?
E136  F1         Ø36Ø          POP     AF
E137  3EØD       Ø361          LD      A,CR             ;YES.  CONVERT
E139  F5         Ø362          PUSH    AF               ;'<' TO A CR.
E13A  D5         Ø363          PUSH    DE
E13B  C5         Ø364          PUSH    BC
E13C  CDC2EØ     Ø365          CALL    CMND1
E13F  C1         Ø366          POP     BC
E14Ø  D1         Ø367          POP     DE
E141  F1         Ø368 PC3:     POP     AF
E142  E1         Ø369          POP     HL
E143  C9         Ø37Ø          RET
                 Ø371 ;
                 Ø372 ;
                 Ø373 ; GET CHARACTER. RETURNS IT IN A.
                 Ø374 ; ALTERS F.
                 Ø375 ;
E144  CD16EØ     Ø376 GCHR:    CALL    GBYTE
E147  CD12E1     Ø377          CALL    PCHR
E14A  28F8       Ø378          JR      Z,GCHR           ;IF NULL DON'T RETURN
E14C  C9         Ø379          RET
                 Ø38Ø ;
                 Ø381 ;
                 Ø382 ; CRLF. ALTERS A ONLY.
                 Ø383 ;
E14D  3EØD       Ø384 CRLF:    LD      A,CR
E14F  18C1       Ø385          JR      PCHR
                 Ø386 ;
                 Ø387 ;
                 Ø388 ; LOADS HL WITH SOURCE ADDR, BC & DE
                 Ø389 ; WITH THE INCREMENT.  ENDS WITH A CRLF.
                 Ø39Ø ;
E151  97         Ø391 L2NCRØ:  SUB     A
                 Ø392 ;
E152  CD8BE1     Ø393 L2NCR:   CALL    LD2N
                 Ø394 ;
                 Ø395 ; SKIP INITIAL SPACES.
                 Ø396 ; IF DELIMITER NOT A CR, ERROR
                 Ø397 ;
E155  CDDEE1     Ø398 SKSGCR:  CALL    SKSG             ;WAIT FOR NON-SPACE
E158  2Ø9B       Ø399          JR      NZ,ERRV1         ;IF NOT CR, ERROR
E15A  EB         Ø4ØØ          EX      DE,HL
```

```
E15B  C9                        RET
                        ;
                        ;
                        ; PRINT THE NUMBER IN HL, FOLLOWED BY A COLON.
                        ; PRESERVES ALL REGISTERS EXCEPT A.
                        ;
E15C  CD4DE1            PCADDR: CALL    CRLF
                        ;
E15F  CDF2E1            PADDR:  CALL    PNHL
E162  3E3A                      LD      A,':'
E164  18AC                      JR      PCHR
                        ;
                        ;
                        ; COMMAND
                        ;
E166  CDA5E1            VERIF:  CALL    L3NCR           ;GET 3 OPERANDS
                        ;
                        ; COMPARES TWO AREAS OF MEMORY.  ENTER WITH
                        ; SOURCE IN HL, DESTINATION IN DE & COUNT
                        ; IN BC.  ALTERS ALL REGISTERS.
                        ;
E169  1A                VRFY:   LD      A,(DE)
E16A  EDA1                      CPI                     ;COMPARE TO SOURCE
E16C  2B                        DEC     HL
E16D  C4F2E1                    CALL    NZ,PNHL         ;PRINT SOURCE ADDR
E170  C4E9E1                    CALL    NZ,PSNM         ; & CONTENTS
E173  EB                        EX      DE,HL
E174  C4E9E1                    CALL    NZ,PSNM         ; & DEST CONTENTS
E177  C4EFE1                    CALL    NZ,PSNHL        ; & DEST ADDR
E17A  C44DE1                    CALL    NZ,CRLF
E17D  EB                        EX      DE,HL
E17E  23                        INC     HL
E17F  13                        INC     DE
E180  E0                        RET     PO              ;IF BC=0, DONE.
E181  18E6                      JR      VRFY
                        ;
                        ;
                        ; COMMAND
                        ;
E183  CDA5E1            MOVE:   CALL    L3NCR           ;OPERANDS
E186  CD1AE2                    CALL    MVE             ;MOVE IT
E189  18DE                      JR      VRFY
                        ;
                        ;
                        ;
                        ; LOAD TWO NUMBERS. LOADS DE WITH THE BEGINNING
                        ; ADDR, N1. LOADS BC & HL WITH THE INCREMENT
                        ; N2-N1+1 (OR WITH N2 IF THE OPR IS 'S').
                        ; RETURNS WITH LAST DELIMITER IN A.
                        ;
                        ;
E18B  CDAEE1            LD2N:   CALL    GNHL            ;N1 TO HL, DELIM TO A
E18E  EB                        EX      DE,HL           ;SAVE N1 IN DE
E18F  CDDEE1                    CALL    SKSG            ;GET NEXT NON-SPACE
E192  FE53                      CP      'S'+CASE        ;SWATH?
E194  2005                      JR      NZ,L2N1
                        ;
```

```
E196  CDADE1              CALL    GNHLØ           ;YES. INCREMENT TO HL.
E199  18Ø7                JR      L2N2
                  ;
E19B  CDAEE1      L2N1:   CALL    GNHL            ;INCREMENT
E19E  B7                  OR      A               ;CLEAR CY
E19F  ED52                SBC     HL,DE           ;N2-N1
E1A1  23                  INC     HL              ;INCLUDE END POINT
E1A2  44          L2N2:   LD      B,H
E1A3  4D                  LD      C,L             ;BC GETS THE INCRM
E1A4  C9                  RET
                  ;
                  ;
                  ; LOAD 3 OPERANDS. HL GETS THE SOURCE, BC
                  ; THE INCREMENT, AND DE THE 3RD OPERAND.
                  ;
E1A5  CD8BE1      L3NCR:  CALL    LD2N
                  ; (CONTINUE BELOW)
                  ;
                  ;
                  ; ENTER WITH SPACE OR THE FIRST DIGIT
                  ; OF A NUMBER IN A. LOADS HL WITH
                  ; WITH A NEW NUMBER & THEN EXCHANGES
                  ; DE & HL. FINISHES WITH A CRLF.
                  ;
E1A8  CDAEE1      L1NCR:  CALL    GNHL            ;SKIP SPACES, LOAD HL
E1AB  18A8                JR      SKSGCR          ;WAIT FOR A CR
                  ;
                  ;
                  ; CLEARS HL. IF ENTERED WITH HEX CHAR IN A,
                  ; SHIFTS IT INTO HL. O/W, IGNORES LEADING
                  ; SPACES. FIRST CHAR MUST BE HEX. CONTINUES
                  ; SHIFT UNTIL A NON-HEX CHAR RECEIVED & THEN
                  ; RETURNS WITH THE LATTER IN A.
                  ; PRESERVES B,C,D,E.
                  ;
                  ;
E1AD  97          GNHLØ:  SUB     A
                  ;
E1AE  C5          GNHL:   PUSH    BC              ;SAVE
E1AF  21ØØØØ              LD      HL,Ø            ;CLR BUFFER
                  ; STRIP LEADING SPACES & GET CHAR
E1B2  CDDEE1              CALL    SKSG
                  ; FIRST CHAR MUST BE HEX
E1B5  CDC6E1              CALL    HEXSH           ;IF HEX, SHIFT INTO HL
E1B8  DAE5EØ              JP      C,ERROR         ;O/W, ERROR
E1BB  CD44E1      GN1:    CALL    GCHR
E1BE  CDC6E1              CALL    HEXSH           ;IF HEX SHIFT INTO HL
E1C1  78                  LD      A,B             ;RESTORE CHAR
E1C2  3ØF7                JR      NC,GN1  ;IF HEX, CONTINUE
E1C4  C1                  POP     BC              ;IF NON-HEX, DONE
E1C5  C9                  RET
                  ;
                  ;
                  ; IF A CONTAINS HEX CHAR, SHIFTS BINARY EQUIVALENT
                  ; INTO HL. IF NOT HEX, RET WITH CY SET. SAVES
                  ; ORIGINAL CHAR IN B
                  ;
```

```
E1C6   47          Ø515 HEXSH:  LD      B,A
E1C7   D63Ø        Ø516         SUB     'Ø'             ; < 'Ø'?
E1C9   D8          Ø517         RET     C
E1CA   C6E9        Ø518         ADD     'Ø'-['G'+CASE]
E1CC   D8          Ø519         RET     C
E1CD   D6FA        Ø52Ø         SUB     'A'-'G'
E1CF   3ØØ3        Ø521         JR      NC,HX1          ;OK IF >= 'A'
E1D1   C6Ø7        Ø522         ADD     ['A'+CASE]-['9'+1]
E1D3   D8          Ø523         RET     C
E1D4   C6ØA        Ø524 HX1:    ADD     '9'+1-'Ø'
                   Ø525 ; THE A-REG NOW CONTAINS THE HEX DIGIT IN BINARY.
                   Ø526 ; (THE HIGH-ORDER NIBBLE OF A IS Ø.)
E1D6   29          Ø527 HXSH4:  ADD     HL,HL           ;SHIFT 4 BITS INTO HL
E1D7   29          Ø528         ADD     HL,HL
E1D8   29          Ø529         ADD     HL,HL
E1D9   29          Ø53Ø         ADD     HL,HL
E1DA   B5          Ø531         OR      L
E1DB   6F          Ø532         LD      L,A
E1DC   C9          Ø533         RET
                   Ø534 ;
                   Ø535 ;
                   Ø536 ; RETURNS WITH A NON-SPACE IN THE A-REG.
                   Ø537 ; IF ENTERED WITH A-REG CONTAINING A NULL
                   Ø538 ; OR A SPACE, GETS NEW CHARS UNTIL FIRST
                   Ø539 ; NON-SPACE OCCURS. ALTERS AF.
                   Ø54Ø ;
E1DD   97          Ø541 SKSGØ:  SUB     A
                   Ø542 ;
E1DE   B7          Ø543 SKSG:   OR      A               ;DOES A CONTAIN NULL?
E1DF   CC44E1      Ø544 SK1:    CALL    Z,GCHR
E1E2   FE2Ø        Ø545         CP      2ØH             ;SPACE?
E1E4   28F9        Ø546         JR      Z,SK1
E1E6   FEØD        Ø547         CP      CR
E1E8   C9          Ø548         RET
                   Ø549 ;
                   Ø55Ø ;
                   Ø551 ;
                   Ø552 ; PRINT SPACE FOLLOWED BY THE NUMBER POINTED
                   Ø553 ; TO BY HL. ALTERS A ONLY.
                   Ø554 ;
E1E9   CD1ØE1      Ø555 PSNM:   CALL    SPACE
                   Ø556 ; (CONTINUE BELOW)
                   Ø557 ;
                   Ø558 ; PRINTS THE NUMBER POINTED TO BY HL.
                   Ø559 ; PRESERVES ALL REGISTERS BUT A.
                   Ø56Ø ;
E1EC   7E          Ø561 PNM:    LD      A,(HL)
E1ED   18Ø8        Ø562         JR      P2HEX
                   Ø563 ;
                   Ø564 ;
                   Ø565 ;
                   Ø566 ; PRINT THE NUMBER IN HL.
                   Ø567 ; PRESERVES ALL BUT A.
                   Ø568 ;
E1EF   CD1ØE1      Ø569 PSNHL:  CALL    SPACE
                   Ø57Ø ;
E1F2   7C          Ø571 PNHL:   LD      A,H
```

```
E1F3  CDF7E1              CALL    P2HEX
E1F6  7D                  LD      A,L
                ;                                 ;(CONTINUE BELOW)
                ;
                ; PRINT THE NUMBER IN THE A-REGISTER.
                ; PRESERVES ALL REGISTERS.
                ;
E1F7  CDFBE1    P2HEX:    CALL    P1HEX
E1FA  1F                  RRA
E1FB  1F        P1HEX:    RRA
E1FC  1F                  RRA
E1FD  1F                  RRA
E1FE  1F                  RRA
E1FF  F5                  PUSH    AF
E2ØØ  E6ØF                AND     ØFH             ;MASK
E2Ø2  FEØA                CP      1ØD             ; <= 9?
E2Ø4  38Ø2                JR      C,PH1
E2Ø6  C6Ø7                ADD     7               ;A THRU F
E2Ø8  C63Ø      PH1:      ADD     3ØH             ;ASCII BIAS
E2ØA  CD12E1              CALL    PCHR            ;PRINT IT
E2ØD  F1                  POP     AF
E2ØE  C9                  RET
                ;
                ;
                ; PRINT MESSAGE. ENTER WITH ADDR OF MSG
                ; IN HL.  THE MESSAGE IS TERMINATED
                ; AFTER PRINTING A CHARACTER WHOSE
                ; PARITY BIT WAS SET.
                ; PRESERVES FLAGS, INCREMENTS HL.
                ;
                ;
                ;
E2ØF  F5        PMSG:     PUSH    AF              ;SAVE
E21Ø  7E        PS1:      LD      A,(HL)
E211  23                  INC     HL
E212  CD12E1              CALL    PCHR
E215  17                  RLA                     ;LAST CHARACTER?
E216  3ØF8                JR      NC,PS1          ;IF NOT, LOOP
E218  F1                  POP     AF
E219  C9                  RET
                ;
                ;
                ; MOVE FROM ONE LOCATION TO ANOTHER. ENTER
                ; WITH SOURCE ADDR IN HL, DEST IN DE, BYTE
                ; COUNT IN BC.  PRESERVES ALL REGISTERS.
                ;
E21A  E5        MVE:      PUSH    HL              ;SOURCE
E21B  D5                  PUSH    DE              ;DEST
E21C  C5                  PUSH    BC              ;BYTE COUNT
E21D  EDBØ                LDIR
E21F  C1                  POP     BC
E22Ø  D1                  POP     DE
E221  E1                  POP     HL
E222  C9                  RET
                ;
                ;
                ; COMMAND
```

```
                    Ø629 ;
                    Ø63Ø ; GO <CR> EXECUTION BEGINS AT USER PC.
                    Ø631 ;
                    Ø632 ; COMMAND
                    Ø633 ;
                    Ø634 ; GO <ADDR1>/<ADDR2> ... <ADDRN>
                    Ø635 ; EXECUTION BEGINS AT ADDR1 WITH BREAKPOINTS SET
                    Ø636 ; AT ADDR2,...,ADDRN.
                    Ø637 ;
                    Ø638 GO:
                    Ø639 ; B GETS NBRKPT+1 (MAX. NUMBER OF BP + 1)
                    Ø64Ø ; C, THE BREAKPOINT FLAG, GETS Ø (NO BP SET)
E223  Ø1ØØØ6        Ø641         LD      BC,[[NBRKPT+1] SHL 8]+Ø
E226  CDDEE1        Ø642 GO1:    CALL    SKSG            ;WAIT FOR NON-SPACE
E229  283A          Ø643         JR      Z,RETN          ;RETN IF CR
E22B  FE2F          Ø644         CP      '/'             ;BP?
E22D  2ØØD          Ø645         JR      NZ,GO3
E22F  4F            Ø646         LD      C,A             ;SET BRKPT FLAG (2FH)
E23Ø  213ØØØ        Ø647         LD      HL,RSTLC        ;TRANSFER
E233  36C3          Ø648         LD      (HL),ØC3H       ;'JP SVMS' TO
E235  2145EØ        Ø649         LD      HL,SVMS
E238  2231ØØ        Ø65Ø         LD      (RSTLC+1),HL    ;RST LOC
E23B  97            Ø651         SUB     A
E23C  CDAEE1        Ø652 GO3:    CALL    GNHL            ;GET ADDR
E23F  CB69          Ø653         BIT     5,C             ; FLAG SET?
E241  EB            Ø654         EX      DE,HL
E242  DDE5          Ø655         PUSH    IX
E244  E1            Ø656         POP     HL
E245  2818          Ø657         JR      Z,GO5           ;JUMP IF NO BP
                    Ø658 ;
E247  Ø5            Ø659         DEC     B               ;IF TOO MANY BP,
E248  CAE5EØ        Ø66Ø         JP      Z,ERROR         ;ERROR.
E24B  6E            Ø661         LD      L,(HL)          ;HL = BPSP
                    Ø662 ;
E24C  23            Ø663         INC     HL              ;BUMP BPSP
E24D  EB            Ø664         EX      DE,HL           ;DE=BPSP, HL= BP ADDR
E24E  EDAØ          Ø665         LDI
E25Ø  2B            Ø666         DEC     HL
E251  36F7          Ø667         LD      (HL),ØC7H+RSTLC ;RST INSTRUCTION
E253  EB            Ø668         EX      DE,HL           ;HL=BPSP
E254  73            Ø669         LD      (HL),E          ;BP ADDR TO STACK
E255  23            Ø67Ø         INC     HL
E256  72            Ø671         LD      (HL),D
E257  23            Ø672         INC     HL
E258  36ØB          Ø673         LD      (HL),BPMRK      ;PUNCTUATION (BP SET)
E25A  DD75ØØ        Ø674         LD      (IX),L
E25D  18C7          Ø675         JR      GO1
                    Ø676 ; CHANGE USER PC
E25F  2B            Ø677 GO5:    DEC     HL
E26Ø  72            Ø678         LD      (HL),D
E261  2B            Ø679         DEC     HL
E262  73            Ø68Ø         LD      (HL),E
E263  18C1          Ø681         JR      GO1             ;BACK FOR MORE
                    Ø682 ;
E265  E1            Ø683 RETN:   POP     HL              ;STRIP ADDR FROM STK
E266  E1            Ø684         POP     HL              ;UHL'
E267  D1            Ø685         POP     DE              ;UDE'
```

```
E268   C1          0686          POP     BC                 ;UBC'
E269   F1          0687          POP     AF                 ;UAF'
E26A   D9          0688          EXX
E26B   08          0689          EX      AF,AF'
                   0690 ;
E26C   F1          0691          POP     AF                 ;UIN
E26D   ED47        0692          LD      I,A                ; UI
E26F   F3          0693          DI
E270   3001        0694          JR      NC,RT1
E272   FB          0695          EI
                   0696 ;IFF NOW RESTORED
E273   FDE1        0697 RT1:     POP     IY                 ;UIY
E275   DDE1        0698          POP     IX                 ;UIX
E277   D1          0699          POP     DE                 ;USP
                   0700 ;
                   0701 ; COPY THE REMAINDER OF THE SYS STACK
                   0702 ; TO THE USER STACK. IF THIS TRANSFER
                   0703 ; IS MADE WITHOUT ERROR, SWITCH TO THE
                   0704 ; USER STACK.  OTHERWISE, RETAIN THE
                   0705 ; SYSTEM STACK.
                   0706 ;
E278   210A00      0707          LD      HL,10D
E27B   45          0708          LD      B,L
E27C   39          0709          ADD     HL,SP
E27D   EB          0710          EX      DE,HL
E27E   1B          0711 RT2:     DEC     DE
E27F   2B          0712          DEC     HL
E280   1A          0713          LD      A,(DE)
E281   77          0714          LD      (HL),A
E282   BE          0715          CP      (HL)
E283   2003        0716          JR      NZ,RT3
E285   10F7        0717          DJNZ    RT2
E287   F9          0718          LD      SP,HL
                   0719 ;
E288   E1          0720 RT3:     POP     HL
E289   D1          0721          POP     DE
E28A   C1          0722          POP     BC
E28B   F1          0723          POP     AF
E28C   C9          0724          RET
                   0725 ;
                   0726 ;
                   0727 ; COMMAND.  DISPLAY REGISTERS.
                   0728 ;
                   0729 ; DR
                   0730 ;
                   0731 ; COMMAND.  DISPLAY MEMORY.
                   0732 ;
                   0733 ; DM <STARTING ADDR> <ENDING ADDR OR SWATH>
                   0734 ;
                   0735 ;
E28D   018041      0736 DISPL:   LD      BC,[['A'+CASE] SHL 8]+80H ;[FOR DR}
E290   203F        0737          JR      NZ,SUBR2           ;IF NOT 'M', DR
                   0738 ;
                   0739 ;
E292   CD51E1      0740 DSPM:    CALL    L2NCR0             ;GET OPERANDS
E295   1610        0741 DSPM1:   LD      D,16               ;BYTE COUNT
E297   CD5CE1      0742          CALL    PCADDR             ;ADDRESS
```

```
E29A  CDE9E1      Ø743 DM2:    CALL    PSNM            ;MEM CONTENTS
E29D  EDA1        Ø744         CPI                     ;INC HL & DEC BC
E29F  E24DE1      Ø745         JP      PO,CRLF
E2A2  15          Ø746         DEC     D
E2A3  28FØ        Ø747         JR      Z,DSPM1
E2A5  7A          Ø748         LD      A,D
E2A6  E6Ø3        Ø749         AND     3
E2A8  CC1ØE1      Ø75Ø         CALL    Z,SPACE
E2AB  CC1ØE1      Ø751         CALL    Z,SPACE
E2AE  18EA        Ø752         JR      DM2
                  Ø753 ;
                  Ø754 ;
                  Ø755 ; COMMAND.  SUBSTITUTE MEMORY LOCATION.
                  Ø756 ;
                  Ø757 ; SM <ADDR>
                  Ø758 ;
                  Ø759 ; COMMAND.  SUBSTITUTE USER-REGISTER.
                  Ø76Ø ;
                  Ø761 ; S<REGISTER NAME>
                  Ø762 ;
                  Ø763 ; REGISTER NAMES: P [PC], S [SP],
                  Ø764 ; A, F, B, C, D, E, H [HL],
                  Ø765 ; I, N [IFF], X [IX], Y [IY],
                  Ø766 ; A',F',B',C',D',E',H' [HL'].
                  Ø767 ;
                  Ø768 ;
E2BØ  2Ø16        Ø769 SUBST:  JR      NZ,SUBR         ;IN NOT 'M', SR
                  Ø77Ø ;
                  Ø771 ;
E2B2  97          Ø772 SUBM:   SUB     A
E2B3  47          Ø773         LD      B,A             ;1-BYTE MASK
E2B4  CDA8E1      Ø774         CALL    L1NCR
E2B7  EB          Ø775         EX      DE,HL           ;HL GETS ADDR
E2B8  CC5CE1      Ø776 SM1:    CALL    Z,PCADDR
E2BB  CC1ØE1      Ø777         CALL    Z,SPACE
                  Ø778 ; PRINT CURRENT VALUE, REQUEST NEW VALUE &
                  Ø779 ; PRINT IT IF GIVEN
E2BE  CD32E3      Ø78Ø         CALL    GSUBV
E2C1  C8          Ø781         RET     Z               ;IF CR, DONE.
E2C2  23          Ø782         INC     HL
E2C3  3EØ7        Ø783         LD      A,7             ;PRINT ADDRESS IF IT
E2C5  A5          Ø784         AND     L               ;IS A MULTIPLE OF 8
E2C6  18FØ        Ø785         JR      SM1
                  Ø786 ;
                  Ø787 ;
E2C8  47          Ø788 SUBR:   LD      B,A
E2C9  CD44E1      Ø789         CALL    GCHR
E2CC  FE27        Ø79Ø         CP      ''''
E2CE  2ØØ2        Ø791         JR      NZ,SR2
E2DØ  ØC          Ø792         INC     C               ;TURN ON THE PRIME-FLAG
E2D1  97          Ø793 SUBR2:  SUB     A
E2D2  CD55E1      Ø794 SR2:    CALL    SKSGCR          ;WAIT FOR CR
E2D5  78          Ø795 SR3:    LD      A,B
E2D6  D641        Ø796         SUB     'A'+CASE        ;CHECK THE RANGE
E2D8  DAE5EØ      Ø797         JP      C,ERROR
E2DB  FE19        Ø798         CP      'Y'-'A'+1
E2DD  D2E5EØ      Ø799         JP      NC,ERROR
```

```
E2EØ  5F                    LD     E,A
E2E1  16ØØ                  LD     D,Ø
E2E3  21D7E3                LD     HL,RGTBL
E2E6  19                    ADD    HL,DE
E2E7  7E                    LD     A,(HL)
E2E8  B7                    OR     A
E2E9  2833                  JR     Z,SR6             ;IF ENTRY = Ø, SKIP
E2EB  1EØØ                  LD     E,Ø
E2ED  CB41                  BIT    Ø,C               ;PRIME?
E2EF  28Ø6                  JR     Z,SR4
E2F1  CB76                  BIT    PF,(HL)           ;YES. PRIMEABLE REG?
E2F3  2829                  JR     Z,SR6             ;IF NOT, SKIP.
E2F5  1E1Ø                  LD     E,DUAF-DUAF2
E2F7  E61F           SR4:   AND    1FH               ;STRIP FLAGS FROM ENTRY
E2F9  83                    ADD    E
E2FA  5F                    LD     E,A
E2FB  C5                    PUSH   BC                ;SAVE
E2FC  78                    LD     A,B               ;PRINT REG NAME
E2FD  CD12E1                CALL   PCHR
E3ØØ  FE48                  CP     'H'+CASE
E3Ø2  3E4C                  LD     A,'L'+CASE
E3Ø4  CC12E1                CALL   Z,PCHR
E3Ø7  EE71                  XOR    'L'+CASE XOR '=';CLEAR CY, A = '='.
E3Ø9  CB41                  BIT    Ø,C               ;PRIME?
E3ØB  28Ø2                  JR     Z,SR5
E3ØD  3E27                  LD     A,''''
E3ØF  CD12E1         SR5:   CALL   PCHR
E312  46                    LD     B,(HL)            ;SAVE ORIGINAL ENTRY
E313  DDE5                  PUSH   IX
E315  E1                    POP    HL                ;STACK FRAME
E316  ED52                  SBC    HL,DE             ;HL -> USER REG
E318  CD32E3                CALL   GSUBV             ;PRINT VALUE, REQUEST NEW
E31B  78                    LD     A,B               ;SAVE
E31C  C1                    POP    BC
E31D  C8                    RET    Z                 ;DONE IF CR
                     ;
E31E  Ø4             SR6:   INC    B                 ;NEXT REG
E31F  Ø7                    RLCA                     ;Y OR H?
E32Ø  3ØB3                  JR     NC,SR3            ;IF NEITHER, LOOP
E322  Ø7                    RLCA                     ;YES, IS IT Y?
E323  CD4DE1         SUBR3: CALL   CRLF              ;[ENTRY FOR DISPLAYING PC
E326  38Ø5                  JR     C,SR8
E328  Ø641                  LD     B,'A'+CASE        ;YES, IT IS Y.
E32A  ØC                    INC    C                 ;TURN ON PRIME-FLAG
E32B  18A8                  JR     SR3
E32D  CB41           SR8:   BIT    Ø,C               ;NO. H OR H'?
E32F  28A4                  JR     Z,SR3             ;IF H, LOOP.
E331  C9                    RET                      ;IT IS H'. DONE.
                     ;
                     ;
                     ; ENTER WITH HL POINTING TO MEMORY &
                     ; B CONTAINING THE 1-BYTE OR 2-BYTE FLAG.
                     ; PRINTS SPACE, CONTENTS OF (HL), & ALSO (HL-1) FOR
                     ; 2-BYTE REGS, GETS SUBSTITUTION VALUE & LOADS IT.
                     ; RETURNS WITH Z-FLAG SET IFF THE DELIMITER IS
                     ; A CARRIAGE-RETURN.
                     ; PRESERVES BC & HL.
```

```
                  ;
E332  CDECE1      GSUBV:  CALL    PNM             ;PRINT (HL)
E335  CB68                BIT     B2F,B           ;2-BYTE REG?
E337  28Ø4                JR      Z,GS1
E339  2B                  DEC     HL
E33A  CDECE1              CALL    PNM             ;LO BYTE
E33D  79          GS1:    LD      A,C             ;SUBST-OR-DISPLAY FLAG
E33E  Ø7                  RLCA
E33F  38ØA                JR      C,GS2           ;IF DISPLAY, EXIT.
E341  3E2E                LD      A,'.'
E343  CD12E1              CALL    PCHR
E346  CD44E1              CALL    GCHR
E349  FE2F                CP      '.'+1           ;SUBSTITUTION?
E34B  DC12E1      GS2:    CALL    C,PCHR          ;IF NOT, PRINT ANOTHER.
E34E  38ØC                JR      C,GS3
E35Ø  EB                  EX      DE,HL
E351  CDAEE1              CALL    GNHL            ;NEW VALUE
E354  EB                  EX      DE,HL
E355  73                  LD      (HL),E
E356  CB68                BIT     B2F,B
E358  28Ø2                JR      Z,GS3
E35A  23                  INC     HL
E35B  72                  LD      (HL),D
E35C  FEØD        GS3:    CP      CR
E35E  C41ØE1              CALL    NZ,SPACE
E361  C9                  RET
                  ;
                  ;
                  ;...SUBDM ØØ 7E 5 585 BY 5 1ØØ DBE++
                  ;
                  ;
                  ; COMMAND
                  ; SELECT UART-A OR UART-B.
                  ;
                  ; UA
                  ; UB
                  ;
E362  CDA8E1      UART:   CALL    L1NCR           ;A OR B?
E365  7B                  LD      A,E
E366  FEØB                CP      ØBH
E368  2ØØ5                JR      NZ,UARTA
E36A  3E8Ø                LD      A,8ØH
E36C  D3Ø4                OUT     APARLP,A
E36E  C9                  RET
                  ;
E36F  97          UARTA:  SUB     A
E37Ø  D354                OUT     BPARLP,A
E372  C9                  RET
                  ;
                  ;
                  ; COMMAND
                  ; READ BINARY INPUT FROM DATA PORT
                  ;
E373  CD52E1      READB:  CALL    L2NCR           ;GET MEM ADDRS
E376  CDØEEØ      RB1:    CALL    CHKIN           ;GET INPUT
E379  28FB                JR      Z,RB1
E37B  77                  LD      (HL),A          ;TO MEM
```

```
E37C  EDA1                   CPI
E37E  EØ                     RET     PO
E37F  18F5                   JR      RB1
                   ;
                   ;
                   ; COMMAND
                   ; WRITE BINARY OUTPUT TO DATA PORT
                   ;
E381  CD52E1       WRITB:    CALL    L2NCR           ;GET MEM ADDRS
E384  7E           WB1:      LD      A,(HL)
E385  CD1EEØ                 CALL    PBYTE
E388  EDA1                   CPI
E38A  EØ                     RET     PO
E38B  18F7                   JR      WB1
                   ;
                   ;
                   ; COMMAND
                   ; PRINT NULLS ON THE CURRENT DEVICE.
                   ;
                   ; N <NUMBER-OF-NULLS>
                   ;
E38D  CDA8E1       NULLS:    CALL    L1NCR
E39Ø  43                     LD      B,E
E391  97                     SUB     A
E392  CD12E1       N2:       CALL    PCHR
E395  1ØFB                   DJNZ    N2
E397  C9                     RET
                   ;
                   ;
                   ; COMMAND
                   ; OUT <DATA-BYTE> <PORT NNUMBER>
                   ;
E398  CDAEE1       OUTP:     CALL    GNHL
E39B  EB                     EX      DE,HL           ;E GETS DATA
E39C  CDA8E1                 CALL    L1NCR           ;GET PORT NUMBER
                   ;
E39F  4B                     LD      C,E             ; TO C
E3AØ  ED69                   OUT     (C),L
E3A2  C9                     RET
                   ;
                   ;
                   ; BAUD RATES.
                   ; WITH THE CROMEMCO TUART: 192ØØ, 96ØØ, 48ØØ,
                   ;                    24ØØ, 12ØØ, 3ØØ, 15Ø, 11Ø.
                   ;
                   ; WITH THE 3P+S: 24ØØ, 3ØØ, 11Ø.
                   ;
                   ;
E3A3  94CEA292     BAUDRS: DB        94H,ØCEH,ØA2H,92H,88H,84H,82H,1
      88848201
                   ;
                   ;
E3AB  ØAØØ8Ø       LFNN:     DB      LF,Ø,Ø OR 8ØH
                   ;
                   ;
E3AE  BA           PRMPT:    DB      ':' OR 8ØH
                   ; THE COMMAND TBL MUST IMMEDIATELY FOLLOW
```

```
                ; THE PROMPT MESSAGE
E3AF  8DE2              DW      DISPL            ;DISPLAY: DM, DR
E3B1  E5EØ              DW      ERROR            ;E
E3B3  E5EØ              DW      ERROR            ;F
E3B5  23E2              DW      GO               ;GO; GO/WITH BREAKPOINTS
E3B7  E5EØ              DW      ERROR            ;H
E3B9  2DEØ              DW      INITBAUD         ;INITIALIZE BAUD RATE
E3BB  E5EØ              DW      ERROR            ;J
E3BD  E5EØ              DW      ERROR            ;K
E3BF  E5EØ              DW      ERROR            ;L
E3C1  83E1              DW      MOVE             ;MOVE A BLOCK OF MEMORY
E3C3  8DE3              DW      NULLS            ;NULLS
E3C5  98E3              DW      OUTP             ;OUTPUT
E3C7  ECEØ              DW      PROG             ;PROGRAM
E3C9  E5EØ              DW      ERROR            ;Q
E3CB  73E3              DW      READB            ;READ BINARY OR ASCII
E3CD  BØE2              DW      SUBST            ;SUBSTITUTE: SM, SA, SB,
E3CF  E5EØ              DW      ERROR            ;T
E3D1  62E3              DW      UART             ;UART: UA, UB
E3D3  66E1              DW      VERIF            ;VERIFY BLOCKS OF MEMORY
E3D5  81E3              DW      WRITB            ;WRITE BINARY OR ASCII
                ;
      (ØØ4Ø)    PM:     EQU     1 SHL PF         ;PRIMEABLE-REG MASK
      (ØØØØ)    B1M:    EQU     Ø                ;1-BYTE REG MASK
      (ØØ2Ø)    B2M:    EQU     1 SHL B2F        ;2-BYTE REG MSK
      (ØØ8Ø)    CRM:    EQU     1 SHL CRF        ;CARRIAGE-RETURN MSK
                ;
E3D7  43        RGTBL:  DB      -DUAF OR PM      ;A
E3D8  45                DB      -DUBC OR PM      ;B
E3D9  46                DB      -DUBC+1 OR PM    ;C
E3DA  47                DB      -DUDE OR PM      ;D
E3DB  48                DB      -DUDE+1 OR PM    ;E
E3DC  44                DB      -DUAF+1 OR PM    ;F
E3DD  ØØ                DB      Ø
E3DE  E9                DB      -DUHL OR PM OR B2M OR CRM ;H [HL]
E3DF  11                DB      -DUIN OR B1M     ;I
E3EØ  ØØ                DB      Ø
E3E1  ØØ                DB      Ø
E3E2  ØØ                DB      Ø
E3E3  ØØ                DB      Ø
E3E4  12                DB      -DUIN+1 OR B1M   ;N [INTERRUPT FF]
E3E5  ØØ                DB      Ø
E3E6  21                DB      -DUPC OR B2M     ;PC
E3E7  ØØ                DB      Ø
E3E8  ØØ                DB      Ø
E3E9  2B                DB      -DUSP OR B2M     ;SP
E3EA  ØØ                DB      Ø
E3EB  ØØ                DB      Ø
E3EC  ØØ                DB      Ø
E3ED  ØØ                DB      Ø
E3EE  2D                DB      -DUIX OR B2M     ;X [IX]
E3EF  AF                DB      -DUIY OR B2M OR CRM ;Y [IY]
                ;
                ;
E3FØ  ØDØD4352  HEAD:   DB      CR,CR,'CROMEMCO ZM1.','4' OR 8ØH
      4F4D454D
      434F2Ø5A
```

```
        4D312EB4
                     1Ø25 ;


  Errors             Ø
  Range Count        Ø
```

Symbol Table

```
ABAUDP  ØØØØ  ACMNDP  ØØØ2  ALT     ØØ7D  APARLP  ØØØ4  B1M     ØØØØ
B2F     ØØØ5  B2M     ØØ2Ø  BAUDRS  E3A3  BCMNDP  ØØ52  BPARLP  ØØ54
BPMRK   ØØØB  BPSTOR  ØØ16  CASE    ØØØØ  CHKIN   EØØE  CL1     EØA4
CL2     EØB3  CLBP    EØAØ  CMND    EØBE  CMND1   EØC2  COM1    EØ51
COM3    EØ82  COM4    EØ8B  COMMON  EØ4A  CR      ØØØD  CRF     ØØØ7
CRLF    E14D  CRM     ØØ8Ø  CSTART  EØØØ  DATA    ØØØ1  DAV     ØØ4Ø
DISPL   E28D  DM2     E29A  DSPM    E292  DSPM1   E295  DUAF    FFFD
DUAF2   FFED  DUBC    FFFB  DUBC2   FFEB  DUDE    FFF9  DUDE2   FFE9
DUHL    FFF7  DUHL2   FFE7  DUIN    FFEF  DUIX    FFF3  DUIY    FFF1
DUPC    FFFF  DUSP    FFF5  ERROR   EØE5  ERRV1   EØF5  ESC     ØØ1B
ESCPE   EØEA  GBYTE   EØ16  GCHR    E144  GN1     E1BB  GNHL    E1AE
GNHLØ   E1AD  GO      E223  GO1     E226  GO3     E23C  GO5     E25F
GS1     E33D  GS2     E34B  GS3     E35C  GSUBV   E332  HEAD    E3FØ
HEXSH   E1C6  HX1     E1D4  HXSH4   E1D6  INIT    EØ29  INITBA  EØ2D
IT1     EØ34  L1NCR   E1A8  L2N1    E19B  L2N2    E1A2  L2NCR   E152
L2NCRØ  E151  L3NCR   E1A5  LD2N    E18B  LENRGS  ØØ1A  LF      ØØØA
LFNN    E3AB  MOVE    E183  MVE     E21A  N2      E392  NBRKPT  ØØØ5
NULLS   E38D  OUTP    E398  P1HEX   E1FB  P2HEX   E1F7  P2NMS   E1Ø8
PADDR   E15F  PBY1    EØ1F  PBYTE   EØ1E  PC1     E113  PC2     E122
PC3     E141  PCADDR  E15C  PCHR    E112  PF      ØØØ6  PH1     E2Ø8
PM      ØØ4Ø  PMSG    E2ØF  PNHL    E1F2  PNM     E1EC  PR1     EØFB
PRMPT   E3AE  PROG    EØEC  PS1     E21Ø  PSNHL   E1EF  PSNM    E1E9
RB1     E376  READB   E373  RETN    E265  RGTBL   E3D7  RSTLC   ØØ3Ø
RT1     E273  RT2     E27E  RT3     E288  SK1     E1DF  SKSG    E1DE
SKSGØ   E1DD  SKSGCR  E155  SM1     E2B8  SPACE   E11Ø  SR2     E2D2
SR3     E2D5  SR4     E2F7  SR5     E3ØF  SR6     E31E  SR8     E32D
STAT    ØØØØ  SUBM    E2B2  SUBR    E2C8  SUBR2   E2D1  SUBR3   E323
SUBST   E2BØ  SVMS    EØ45  TBE     ØØ8Ø  TEMPS   ØØ16  UART    E362
UARTA   E36F  VERIF   E166  VRFY    E169  WB1     E384  WRITB   E381
WSTART  EØØ8
```

Cross Reference

```
ABAUDP  ØØØ7  Ø131
ACMNDP  ØØØ6  Ø133
ALT     ØØ28  Ø345
APARLP  ØØØ8  Ø899
B1M     Ø993  1ØØ5 1Ø1Ø
B2F     ØØ21  Ø859 Ø876 Ø994
B2M     Ø994  1ØØ4 1Ø12 1Ø15 1Ø2Ø 1Ø21
BAUDRS  Ø962  Ø13Ø
BCMNDP  ØØØ9  Ø111
BPARLP  ØØ1Ø  Ø11Ø Ø9Ø3
BPMRK   ØØ17  Ø233 Ø673
BPSTOR  ØØ15  ØØ16
CASE    ØØ2Ø  Ø221 Ø269 Ø285 Ø455 Ø518 Ø522 Ø736 Ø796 Ø819 Ø82Ø Ø822 Ø842
CHKIN   ØØ81  ØØ9Ø Ø347 Ø911
CL1     Ø232  Ø243
CL2     Ø245  Ø234
CLBP    Ø228  Ø294
CMND    Ø258  Ø258
CMND1   Ø26Ø  Ø365
COM1    Ø164  Ø168
COM3    Ø2Ø3  Ø194
COM4    Ø2Ø8  Ø2Ø6
COMMON  Ø154  ØØ65 ØØ76
CR      ØØ25  Ø137 Ø356 Ø361 Ø384 Ø547 Ø88Ø 1Ø24 1Ø24
CRF     ØØ23  Ø995
CRLF    Ø384  Ø257 Ø4Ø7 Ø43Ø Ø745 Ø84Ø
CRM     Ø995  1ØØ4 1Ø21
CSTART  ØØ61
DATA    ØØØ5  ØØ84 Ø1Ø3
DAV     ØØ11  ØØ82
DISPL   Ø736  Ø971
DM2     Ø743  Ø752
DSPM    Ø74Ø
DSPM1   Ø741  Ø747
DUAF    ØØ34  Ø812 Ø997 1ØØ2
DUAF2   ØØ42  Ø812
DUBC    ØØ35  Ø998 Ø999
DUBC2   ØØ43
DUDE    ØØ36  1ØØØ 1ØØ1
DUDE2   ØØ44
DUHL    ØØ37  Ø182 1ØØ4
DUHL2   ØØ45  ØØ47
DUIN    ØØ41  1ØØ5 1Ø1Ø
DUIX    ØØ39  1Ø2Ø
DUIY    ØØ4Ø  1Ø21
DUPC    ØØ33  ØØ47 Ø182 1Ø12
DUSP    ØØ38  1Ø15
ERROR   Ø292  Ø27Ø Ø272 Ø3Ø8 Ø5Ø2 Ø66Ø Ø797 Ø799 Ø972 Ø973 Ø975 Ø977 Ø978 Ø979
              Ø984 Ø987
ERRV1   Ø3Ø8  Ø399
ESC     ØØ27  Ø343
ESCPE   Ø294  Ø344 Ø346
GBYTE   ØØ9Ø  ØØ91 Ø135 Ø136 Ø376
GCHR    Ø376  Ø378 Ø5Ø3 Ø544 Ø789 Ø868
GN1     Ø5Ø3  Ø5Ø6
GNHL    Ø496  Ø452 Ø461 Ø482 Ø652 Ø873 Ø946
GNHLØ   Ø494  Ø458
```

```
GO      Ø638  Ø974
GO1     Ø642  Ø675 Ø681
GO3     Ø652  Ø645
GO5     Ø677  Ø657
GS1     Ø863  Ø86Ø
GS2     Ø87Ø  Ø865
GS3     Ø88Ø  Ø871 Ø877
GSUBV   Ø858  Ø78Ø Ø831
HEAD    1Ø24  Ø219
HEXSH   Ø515  Ø5Ø1 Ø5Ø4
HX1     Ø524  Ø521
HXSH4   Ø527
INIT    Ø11Ø  Ø2Ø1
INITBA  Ø13Ø  Ø976
IT1     Ø133  Ø139
L1NCR   Ø482  Ø774 Ø894 Ø935 Ø948
L2N1    Ø461  Ø456
L2N2    Ø465  Ø459
L2NCR   Ø393  Ø91Ø Ø922
L2NCRØ  Ø391  Ø74Ø
L3NCR   Ø473  Ø3Ø2 Ø416 Ø44Ø
LD2N    Ø452  Ø393 Ø473
LENRGS  ØØ47  Ø249
LF      ØØ26  Ø965
LFNN    Ø965  Ø355
MOVE    Ø44Ø  Ø98Ø
MVE     Ø618  Ø313 Ø441
N2      Ø938  Ø939
NBRKPT  ØØ14  ØØ15 Ø641
NULLS   Ø935  Ø981
OUTP    Ø946  Ø982
P1HEX   Ø581  Ø579
P2HEX   Ø579  Ø562 Ø572
P2NMS   Ø327
PADDR   Ø4Ø9
PBY1    ØØ99  Ø1Ø1
PBYTE   ØØ98  Ø354 Ø924
PC1     Ø342  Ø348
PC2     Ø35Ø
PC3     Ø368  Ø359
PCADDR  Ø4Ø7  Ø742 Ø776
PCHR    Ø341  Ø293 Ø377 Ø385 Ø411 Ø591 Ø6Ø7 Ø818 Ø821 Ø826 Ø867 Ø87Ø Ø938
PF      ØØ22  Ø81Ø Ø992
PH1     Ø59Ø  Ø588
PM      Ø992  Ø997 Ø998 Ø999 1ØØØ 1ØØ1 1ØØ2 1ØØ4
PMSG    Ø6Ø4  Ø22Ø Ø261 Ø357
PNHL    Ø571  Ø4Ø9 Ø425
PNM     Ø561  Ø327 Ø329 Ø858 Ø862
PR1     Ø312  Ø318
PRMPT   Ø968  Ø26Ø
PROG    Ø3Ø2  Ø983
PS1     Ø6Ø5  Ø6Ø9
PSNHL   Ø569  Ø429
PSNM    Ø555  Ø426 Ø428 Ø743
RB1     Ø911  Ø912 Ø916
READB   Ø91Ø  Ø985
RETN    Ø683  Ø643
```

```
RGTBL    Ø997   Ø8Ø2
RSTLC    ØØ19   Ø647 Ø65Ø Ø667
RT1      Ø697   Ø694
RT2      Ø711   Ø717
RT3      Ø72Ø   Ø716
SK1      Ø544   Ø546
SKSG     Ø543   Ø398 Ø454 Ø499 Ø642
SKSGØ    Ø541   Ø267 Ø284
SKSGCR   Ø398   Ø483 Ø794
SM1      Ø776   Ø785
SPACE    Ø335   Ø555 Ø569 Ø75Ø Ø751 Ø777 Ø881
SR2      Ø794   Ø791
SR3      Ø795   Ø838 Ø844 Ø846
SR4      Ø813   Ø8Ø9
SR5      Ø826   Ø824
SR6      Ø836   Ø8Ø6 Ø811
SR8      Ø845   Ø841
STAT     ØØØ4   ØØ81 ØØ99
SUBM     Ø772
SUBR     Ø788   Ø769
SUBR2    Ø793   Ø737
SUBR3    Ø84Ø   Ø222
SUBST    Ø769   Ø986
SVMS     Ø146   Ø649
TBE      ØØ12   Ø1ØØ
TEMPS    ØØ16   Ø163
UART     Ø894   Ø988
UARTA    Ø9Ø2   Ø897
VERIF    Ø416   Ø989
VRFY     Ø422   Ø32Ø Ø435 Ø442
WB1      Ø923   Ø927
WRITB    Ø922   Ø99Ø
WSTART   ØØ73
```