```
                            ORG    :DAD4
                      *
                      *
                      *
                      *   ===============
                      *** PRINT ROUTINES ***
                      *   ===============
                      *
                      *
                      ****************
                      * PRINT MESSAGE *
                      ****************
                      *
                      * Prints a message, which may include internal
                      * references to other submessages.
                      *
                      * Entry: Pointer to message in HL.
                      * Exit:  Pointer after string in HL. Other
                      *        registers preserved.
                      *
                      * Message format: A series of bytes:
                      *         00    End of string.
                      *         01-7F Printed character.
                      *         >= 80 bit 14 set:
                      *               bits 0-13 are offset in program of
                      *               a message (char. terminated by a 0).
                      *               bit 14 unset:
                      *               bits 0-13 are offset in program of
                      *               a string (1 byte length + char.).
                      *
DAD4 F5               PMSG    PUSH  PSW
DAD5 7E               PMS10   MOV   A,M       Get character
DAD6 23                       INX   H         Points to next char.
DAD7 B7               PMS15   ORA   A         Check char.
DAD8 CAE4DA                   JZ    :DAE4     If end message
DADB FAE6DA                   JM    :DAE6     If submessage reference
DADE CD60DD                   CALL  :DD60     Print character in A
DAE1 C3D5DA                   JMP   :DAD5     Next
                      *
DAE4 F1               PMS20   POP   PSW       End message
DAE5 C9                       RET

                      * Submessage reference:

DAE6 FEC0             PMS30   CPI   :C0
DAE8 E5                       PUSH  H
DAE9 6E                       MOV   L,M       Lobyte in L
DAEA DAF6DA                   JC    :DAF6     If reference to string
DAED 67                       MOV   H,A       BASE is at location C000
DAEE CDD4DA                   CALL  :DAD4     Print submessage
DAF1 E1               PMS35   POP   H
DAF2 23                       INX   H
DAF3 C3D5DA                   JMP   :DAD5     Next character

                      * String reference:

DAF6 C640             PMS40   ADI   :40       (BASE SHR 8) - #80
DAF8 67                       MOV   H,A       Hibyte in H
DAF9 CD32DB                   CALL  :DB32     Print substring pntd by HL
DAFC C3F1DA                   JMP   :DAF1     Next character
                      *
                      *
```

```
                 ********************************************
                 * PRINT MESSAGE POINTED TO BY NEXT 2 BYTES *
                 ********************************************
                 *
                 * Entry: Top of stack points to the address where
                 *        the address of the message can be found.
                 * Exit:  AFBCDEHL preserved.
                 *        Returnaddress on stack.
                 *
DAFF E3          PMSGR   XTHL             Get pntr from stack
DB00 D5                  PUSH   D
DB01 5E                  MOV    E,M       Get lobyte address
DB02 23                  INX    H
DB03 56                  MOV    D,M       Get hibyte address
DB04 23                  INX    H
DB05 EB                  XCHG             Addr message in HL
DB06 CDD4DA              CALL   :DAD4     Print message
DB09 EB                  XCHG             HL pnts after message pn'tr
DB0A D1                  POP    D
DB0B E3                  XTHL             Restore stack
DB0C C9                  RET
                 *
                 ************************
                 * CURSOR TO NEXT FIELD *
                 ************************
                 *
                 * Part of 'run PRINT' (0E2B3).
                 * Moves the cursor to a new number output field.
                 * The field size is 12 character positions; field
                 * positions: 0,12,24,36,48.
                 *
DB0D EF          PSKP    RST    5         Ask cursor position
DB0E 0C                  DATA   :0C       and size character screen
DB0F 7D                  MOV    A,L       X-coord in L
DB10 FE30                CPI    :30       Already past last field?
DB12 D227DB              JNC    :DB27     Then print car.ret, abort
DB15 D60C        PSK10   SUI    :0C       ) Minus 12 untill underflow
DB17 D215DB              JNC    :DB15     )
DB1A 2F          PSK15   CMA              Restore pos. value
DB1B 3C                  INR    A

                 * Entry from 'SPC' function:

DB1C 57          PSK20   MOV    D,A       Store nr of spaces required
DB1D 3E20        PSK30   MVI    A,:20
DB1F CD60DD              CALL   :DD60     Print space
DB22 15                  DCR    D         Ready?
DB23 C21DDB              JNZ    :DB1D     Next space if not
DB26 C9                  RET
                 *
DB27 C35EDD      PSK40   JMP    :DD5E     Print car.ret
                 *
                 ********************
                 * CURSOR TO TAB(8) *
                 ********************
                 *
                 * Part of 'List current line' (0ECB3).
                 * Moves cursor to column 8 after linenumber.
                 *
                 * Exit: BC preserved. AFDEHL corrupted.
                 *
```

```
                    SCTAB
DB2A EF             PTAB    RST    5          Ask cursor position and
DB2B 0C                     DATA   :0C        size character screen
DB2C 7D                     MOV    A,L        X-coord after linenr in A
DB2D D606                   SUI    :06        Tab must be 8
DB2F C31ADB                 JMP    :DB1A      Print additional spaces
                    *
                    ***************
                    * PRINT STRING *
                    ***************
                    *
                    * Prints a string of characters pointed to by HL.
                    *
                    * Entry: HL points to string.
                    * Exit:  HL points after string.
                    *        Other registers preserved.
                    *
                    * String format:
                    *        1 byte length (0 = no data).
                    *        N bytes data.
                    *
                    SCSTR
DB32 F5             PSTR    PUSH   PSW
DB33 C5                     PUSH   B
DB34 46                     MOV    B,M        String length in B
DB35 23             PST10   INX    H
DB36 78             PST20   MOV    A,B        Get length
DB37 D601                   SUI    :01        Minus 1
DB39 47                     MOV    B,A        Nr. char still to be printed
DB3A 7E                     MOV    A,M        Get char
DB3B D495D6                 CNC    :D695      Print character if not ready
DB3E D235DB                 JNC    :DB35      Next one if not ready
DB41 C1                     POP    B
DB42 F1                     POP    PSW
DB43 C9                     RET
                    *
                    ***********************
                    * PRINT STRING MESSAGE *
                    ***********************
                    *
                    * Prints a string of characters, pointed to by
                    * HL, length in A.
                    *
                    * Entry: HL: Points to string.
                    *        A:  Number of characters.
                    * Exit:  HL: Points after string.
                    *        AFBCDE preserved.
                    *
                    SCSTM
DB44 F5             PSTRM   PUSH   PSW
DB45 C5                     PUSH   B
DB46 47                     MOV    B,A        String length in B
DB47 C336DB                 JMP    :DB36      Print string
                    *
                    *********************
                    * PRINT A HEX NUMBER *
                    *********************
                    *
                    * Converts MACC to hex in DECBUF and print it.
                    *
                    * Exit: HL points after string in DECBUF.
                    *       BCDE preserved. AF corrupted.
```

```
                     *
DB4A CD2DC0     PHEX      CALL   :C02D      Convert MACC for hex output
DB4D 2A33C0     PGP       LHLD   :C033      Get addr DECBUF
DB50 C332DB               JMP    :DB32      Print string in DECBUF
                *
                **************************
                * PRINT A INTEGER NUMBER *
                **************************
                *
                * Prints an integer number from MACC.
                *
DB53 CD5FDB     PINT      CALL   :DB5F      Convert MACC for output
DB56 C34DDB               JMP    :DB4D      Print contents DECBUF
                *
                *******************************
                * PRINT A FLOATING POINT NUMBER *
                *******************************
                *
                * Prints a FPT number from MACC.
                *
DB59 CD9BCE     PFPT      CALL   :CE9B      Convert MACC for output
DB5C C34DDB               JMP    :DB4D      Print contents DECBUF
                *
                ************************************
                * CONVERT MACC FOR FIXED POINT OUTPUT *
                ************************************
                *
                * Places ASCII-equivalent in 00E4-F0, digits before
                * decimal point in 00F1, length in 00E3.
                *
                * Exit: A: Number of digits.
                *       BCDEHL preserved.
                *
DB5F CD27C0     IBCP      CALL   :C027      Convert INT for output
DB62 C5                   PUSH   B
DB63 0600                 MVI    B,:00      Can trim last dec. place
DB65 CD30C0     BPP       CALL   :C030      Tidy up into external form
DB68 C1                   POP    B
DB69 C9                   RET
                *
                ******************************
                * (Not used, replaced by CE9B) *
                ******************************
                *
DB6A 0601       LD216     MVI    B,:01      ) Part of a previous version
DB6C C364DB               JMP    :DB64      )
                *
                *
DB6F                      END

**************************
* S Y M B O L   T A B L E *
**************************

BPP     DB65   IBCP    DB5F   LD216   DB6A   PFPT    DB59
PGP     DB4D   PHEX    DB4A   PINT    DB53   PMS10   DAD5
PMS15   DAD7   PMS20   DAE4   PMS30   DAE6   PMS35   DAF1
PMS40   DAF6   PMSG    DAD4   PMSGR   DAFF   PSK10   DB15
PSK15   DB1A   PSK20   DB1C   PSK30   DB1D   PSK40   DB27
PSKP    DB0D   PST10   DB35   PST20   DB36   PSTR    DB32
PSTRM   DB44   PTAB    DB2A   SCSTM   DB44   SCSTR   DB32
SCTAB   DB2A
```

```
                    ORG    :DB6F
             *
             *
             *
             *********************************
             * STRINGS FOR MACHINE MESSAGES *
             *********************************
             *
             * The machine messages exist partly from strings,
             * partly from subreferences to other strings.
             * The subreferences can be:
             *     - An address where another string can be found.
             *     - An offset with base at C000 to the other
             *       string.
             * The messages are ended with 00.
             * 20 is space, 0D is carriage return.
             *
             RMS01
DB6F 53      MSG01  DATA   :53        S
DB70 4F             DATA   :4F        O
DB71 4D             DATA   :4D        M
DB72 45             DATA   :45        E
DB73 20             DATA   :20
DB74 8CA5           DATA   :8C,:A5    INPUT
DB76 20             DATA   :20
DB77 49             DATA   :49        I
DB78 47             DATA   :47        G
DB79 4E             DATA   :4E        N
DB7A 4F             DATA   :4F        O
DB7B 52             DATA   :52        R
DB7C 45             DATA   :45        E
DB7D 44             DATA   :44        D
DB7E 00             DATA   :00
             *
DB7F 20      MSG02  DATA   :20
DB80 49             DATA   :49        I
DB81 4E             DATA   :4E        N
DB82 20             DATA   :20
DB83 4C      MLINE  DATA   :4C        L
DB84 49             DATA   :49        I
DB85 4E             DATA   :4E        N
DB86 45             DATA   :45        E
DB87 20             DATA   :20
DB88 00             DATA   :00
             *
DB89 DC0D    MSG03  DATA   :DC,:0D    OUT OF
DB8B 8E56           DATA   :8E,:56    SPACE
DB8D 20             DATA   :20
DB8E 8CD2           DATA   :8C,:D2    FOR
DB90 D88D           DATA   :D8,:8D    MODE
DB92 00             DATA   :00
             *
DB93 20      MSG04  DATA   :20
DB94 52             DATA   :52        R
DB95 45             DATA   :45        E
DB96 DBF7           DATA   :DB,:F7    TYPE
DB98 DB83    MLINR  DATA   :DB,:83    LINE
DB9A 0D             DATA   :0D
DB9B 00             DATA   :00
             *
DB9C 0D      MSG15  DATA   :0D
DB9D 53             DATA   :53        S
```

```
DB9E 45                    DATA  :45        E
DB9F 54                    DATA  :54        T
DBA0 20                    DATA  :20
DBA1 52                    DATA  :52        R
DBA2 45                    DATA  :45        E
DBA3 43                    DATA  :43        C
DBA4 4F                    DATA  :4F        O
DBA5 52                    DATA  :52        R
DBA6 44                    DATA  :44        D
DBA7 2C                    DATA  :2C        ,
DBA8 DBB0          MSG05   DATA  :DB,:B0    START TAPE
DBAA 2C                    DATA  :2C        ,
DBAB DBF7                  DATA  :DB,:F7    TYPE
DBAD 8E56                  DATA  :8E,:56    SPACE
DBAF 00                    DATA  :00
                   *
DBB0 53            MSG06   DATA  :53        S
DBB1 54                    DATA  :54        T
DBB2 41                    DATA  :41        A
DBB3 52                    DATA  :52        R
DBB4 54                    DATA  :54        T
DBB5 DBFD                  DATA  :DB,:FD    TAPE
DBB7 00                    DATA  :00
                   *
DBB8 0D            MSG07   DATA  :0D
DBB9 2A                    DATA  :2A        *
DBBA 2A                    DATA  :2A        *
DBBB 2A                    DATA  :2A        *
DBBC DBE0                  DATA  :DB,:E0    BREAK
DBBE 0D                    DATA  :0D
DBBF 00                    DATA  :00
                   *
DBC0 20            MSG17   DATA  :20
DBC1 4F                    DATA  :4F        O
DBC2 4B                    DATA  :4B        K
DBC3 0D                    DATA  :0D
DBC4 00                    DATA  :00
                   *
DBC5 0D            MSG09   DATA  :0D
DBC6 DBE0                  DATA  :DB,:E0    BREAK
DBC8 00                    DATA  :00
                   *
DBC9 8BCE          MSG10   DATA  :8B,:CE    STOP
DBCB 50                    DATA  :50        P
DBCC 45                    DATA  :45        E
DBCD 44                    DATA  :44        D
DBCE 00                    DATA  :00
                   *
DBCF 8BD6          MSG11   DATA  :8B,:D6    END
DBD1 20                    DATA  :20
DBD2 50                    DATA  :50        P
DBD3 52                    DATA  :52        R
DBD4 4F                    DATA  :4F        O
DBD5 47                    DATA  :47        G
DBD6 52                    DATA  :52        R
DBD7 41                    DATA  :41        A
DBD8 4D                    DATA  :4D        M
DBD9 0D                    DATA  :0D
DBDA 00                    DATA  :00
                   *
DBDB 20            MSG14   DATA  :20
DBDC 42                    DATA  :42        B
```

```
DBDD 41                   DATA  :41       A
DBDE 44                   DATA  :44       D
DBDF 00                   DATA  :00
                  *
DBE0 42           MBREAK  DATA  :42       B
DBE1 52                   DATA  :52       R
DBE2 45                   DATA  :45       E
DBE3 41                   DATA  :41       A
DBE4 4B                   DATA  :4B       K
DBE5 00                   DATA  :00
                  *
DBE6 20           MWITHO  DATA  :20
DBE7 57                   DATA  :57       W
DBE8 49                   DATA  :49       I
DBE9 54                   DATA  :54       T
DBEA 48                   DATA  :48       H
DBEB 4F                   DATA  :4F       O
DBEC 55                   DATA  :55       U
DBED 54                   DATA  :54       T
DBEE 20                   DATA  :20
DBEF 00                   DATA  :00
                  *
DBF0 53           MSTRIN  DATA  :53       S
DBF1 54                   DATA  :54       T
DBF2 52                   DATA  :52       R
DBF3 49           MING    DATA  :49       I
DBF4 4E                   DATA  :4E       N
DBF5 47                   DATA  :47       G
DBF6 00                   DATA  :00
                  *
DBF7 54           MTYPE   DATA  :54       T
DBF8 59                   DATA  :59       Y
DBF9 50                   DATA  :50       P
DBFA 45                   DATA  :45       E
DBFB 20                   DATA  :20
DBFC 00                   DATA  :00
                  *
DBFD 20           MTAPE   DATA  :20
DBFE 54                   DATA  :54       T
DBFF 41                   DATA  :41       A
DC00 50                   DATA  :50       P
DC01 45                   DATA  :45       E
DC02 00                   DATA  :00
                  *
DC03 55           MUNDF   DATA  :55       U
DC04 4E                   DATA  :4E       N
DC05 44                   DATA  :44       D
DC06 45                   DATA  :45       E
DC07 46                   DATA  :46       F
DC08 49                   DATA  :49       I
DC09 4E                   DATA  :4E       N
DC0A 45                   DATA  :45       E
DC0B 44                   DATA  :44       D
DC0C 00                   DATA  :00
                  *
DC0D 4F           MOUTOF  DATA  :4F       O
DC0E 55                   DATA  :55       U
DC0F 54                   DATA  :54       T
DC10 20                   DATA  :20
DC11 4F                   DATA  :4F       O
DC12 46                   DATA  :46       F
DC13 20                   DATA  :20
```

```
DC14 00                     DATA   :00
                    *
DC15 20             MERROR  DATA   :20
DC16 45                     DATA   :45         E
DC17 52                     DATA   :52         R
DC18 52                     DATA   :52         R
DC19 4F                     DATA   :4F         O
DC1A 52                     DATA   :52         R
DC1B 00                     DATA   :00
                    *
                    *************************
                    * STRINGS ERROR MESSAGES *
                    *************************
                    *
                    * Comments: See strings machine messages (DB6F).
                    *
DC1C 8CD9           ERMNF   DATA   :8C,:D9     NEXT
DC1E DBE6                   DATA   :DB,:E6     WITHOUT
DC20 8CD2                   DATA   :8C,:D2     FOR
DC22 00                     DATA   :00
                    *
DC23 53             ERMSN   DATA   :53         S
DC24 59                     DATA   :59         Y
DC25 4E                     DATA   :4E         N
DC26 54                     DATA   :54         T
DC27 41                     DATA   :41         A
DC28 58                     DATA   :58         X
DC29 DC15                   DATA   :DC,:15     ERROR
DC2B 00                     DATA   :00
                    *
DC2C 8BE8           ERMRG   DATA   :8B,:E8     RETURN
DC2E DBE6                   DATA   :DB,:E6     WITHOUT
DC30 8C01                   DATA   :8C,:01     GOSUB
DC32 00                     DATA   :00
                    *
DC33 DC0D           ERMOD   DATA   :DC,:0D     OUT OF
DC35 8CAE                   DATA   :8C,:AE     DATA
DC37 00                     DATA   :00
                    *
DC38 53             ERMSO   DATA   :53         S
DC39 54                     DATA   :54         T
DC3A 41                     DATA   :41         A
DC3B 43                     DATA   :43         C
DC3C 4B                     DATA   :4B         K
DC3D 20                     DATA   :20
DC3E 4F             ERMOV   DATA   :4F         O
DC3F 56                     DATA   :56         V
DC40 45                     DATA   :45         E
DC41 52                     DATA   :52         R
DC42 46                     DATA   :46         F
DC43 4C                     DATA   :4C         L
DC44 4F                     DATA   :4F         O
DC45 57                     DATA   :57         W
DC46 00                     DATA   :00
                    *
DC47 DC0D           ERMOM   DATA   :DC,:0D     OUT OF
DC49 4D                     DATA   :4D         M
DC4A 45                     DATA   :45         E
DC4B 4D                     DATA   :4D         M
DC4C 4F                     DATA   :4F         O
DC4D 52                     DATA   :52         R
DC4E 59                     DATA   :59         Y
```

```
DC4F 00                  DATA  :00
                 *
DC50 DC03        ERMUS   DATA  :DC,:03   UNDEFINED
DC52 20                  DATA  :20
DC53 DB53        MLINN   DATA  :DB,:53   LINE
DC55 4E          MNUMBE  DATA  :4E       N
DC56 55                  DATA  :55       U
DC57 4D                  DATA  :4D       M
DC58 42                  DATA  :42       B
DC59 45                  DATA  :45       E
DC5A 52                  DATA  :52       R
DC5B 00                  DATA  :00
                 *
DC5C 53          ERMBS   DATA  :53       S
DC5D 55                  DATA  :55       U
DC5E 42                  DATA  :42       B
DC5F 53                  DATA  :53       S
DC60 43                  DATA  :43       C
DC61 52                  DATA  :52       R
DC62 49                  DATA  :49       I
DC63 50                  DATA  :50       P
DC64 54                  DATA  :54       T
DC65 DC15                DATA  :DC,:15   ERROR
DC67 00                  DATA  :00
                 *
DC68 44          ERMDO   DATA  :44       D
DC69 49                  DATA  :49       I
DC6A 56                  DATA  :56       V
DC6B 49                  DATA  :49       I
DC6C 53                  DATA  :53       S
DC6D 49                  DATA  :49       I
DC6E 4F                  DATA  :4F       O
DC6F 4E                  DATA  :4E       N
DC70 20                  DATA  :20
DC71 42                  DATA  :42       B
DC72 59                  DATA  :59       Y
DC73 20                  DATA  :20
DC74 5A                  DATA  :5A       Z
DC75 45                  DATA  :45       E
DC76 52                  DATA  :52       R
DC77 4F                  DATA  :4F       O
DC78 00                  DATA  :00
                 *
DC79 43          ERMID   DATA  :43       C
DC7A 4F                  DATA  :4F       O
DC7B 4D                  DATA  :4D       M
DC7C 4D                  DATA  :4D       M
DC7D 41                  DATA  :41       A
DC7E 4E                  DATA  :4E       N
DC7F 44                  DATA  :44       D
DC80 20                  DATA  :20
DC81 49          MINVAL  DATA  :49       I
DC82 4E                  DATA  :4E       N
DC83 56                  DATA  :56       V
DC84 41                  DATA  :41       A
DC85 4C                  DATA  :4C       L
DC86 49                  DATA  :49       I
DC87 44                  DATA  :44       D
DC88 20                  DATA  :20
DC89 00                  DATA  :00
                 *
DC8A DBF7        ERMTM   DATA  :DB,:F7   TYPE
```

```
DC8C 4D                     DATA   :4D        M
DC8D 49                     DATA   :49        I
DC8E 53                     DATA   :53        S
DC8F 4D                     DATA   :4D        M
DC90 41                     DATA   :41        A
DC91 54                     DATA   :54        T
DC92 43                     DATA   :43        C
DC93 48                     DATA   :48        H
DC94 00                     DATA   :00
                  *
DC95 DC0D         ERMOS     DATA   :DC,:0D    OUT OF
DC97 DBF0                   DATA   :DB,:F0    STRING
DC99 20                     DATA   :20
DC9A 8E56                   DATA   :8E,:56    SPACE
DC9C 00                     DATA   :00
                  *
DC9D DBF0         ERMLS     DATA   :DB,:F0    STRING
DC9F 20                     DATA   :20
DCA0 54                     DATA   :54        T
DCA1 4F                     DATA   :4F        O
DCA2 4F                     DATA   :4F        O
DCA3 20                     DATA   :20
DCA4 4C                     DATA   :4C        L
DCA5 4F                     DATA   :4F        O
DCA6 4E                     DATA   :4E        N
DCA7 47                     DATA   :47        G
DCA8 00                     DATA   :00
                  *
DCA9 43           ERMCN     DATA   :43        C
DCAA 41                     DATA   :41        A
DCAB 4E                     DATA   :4E        N
DCAC 27                     DATA   :27        '
DCAD 54                     DATA   :54        T
DCAE 20                     DATA   :20
DCAF 8BC6                   DATA   :8B,:C6    CONT
DCB1 00                     DATA   :00
                  *
DCB2 DC81         ERMIN     DATA   :DC,:81    INVALID
DCB4 DC55                   DATA   :DC,:55    NUMBER
DCB6 00                     DATA   :00
                  *
DCB7 4F           ERMOF     DATA   :4F        O
DCB8 46                     DATA   :46        F
DCB9 46                     DATA   :46        F
DCBA 20                     DATA   :20
DCBB 53                     DATA   :53        S
DCBC 43                     DATA   :43        C
DCBD 52                     DATA   :52        R
DCBE 45                     DATA   :45        E
DCBF 45                     DATA   :45        E
DCC0 4E                     DATA   :4E        N
DCC1 00                     DATA   :00
                  *
DCC2 43           ERMNC     DATA   :43        C
DCC3 4F                     DATA   :4F        O
DCC4 4C                     DATA   :4C        L
DCC5 4F                     DATA   :4F        O
DCC6 52                     DATA   :52        R
DCC7 20                     DATA   :20
DCC8 4E                     DATA   :4E        N
DCC9 4F                     DATA   :4F        O
DCCA 54                     DATA   :54        T
```

```
DCCB 20                    DATA   :20
DCCC 41                    DATA   :41         A
DCCD 56                    DATA   :56         V
DCCE 41                    DATA   :41         A
DCCF 49                    DATA   :49         I
DCD0 4C                    DATA   :4C         L
DCD1 41                    DATA   :41         A
DCD2 42                    DATA   :42         B
DCD3 4C                    DATA   :4C         L
DCD4 45                    DATA   :45         E
DCD5 00                    DATA   :00
                    *
DCD6 DB83           ERMLN  DATA   :DB,:83     LINE
DCD8 DC55           ERMNA  DATA   :DC,:55     NUMBER
DCDA 20                    DATA   :20
DCDB DC0D           MSGOR  DATA   :DC,:0D     OUT OF
DCDD 52                    DATA   :52         R
DCDE 41                    DATA   :41         A
DCDF 4E                    DATA   :4E         N
DCE0 47                    DATA   :47         G
DCE1 45                    DATA   :45         E
DCE2 00                    DATA   :00
                    *
DCE3 DB83           ERMTC  DATA   :DB,:83     LINE
DCE5 54                    DATA   :54         T
DCE6 4F                    DATA   :4F         O
DCE7 4F                    DATA   :4F         O
DCE8 20                    DATA   :20
DCE9 43                    DATA   :43         C
DCEA 4F                    DATA   :4F         O
DCEB 4D                    DATA   :4D         M
DCEC 50                    DATA   :50         P
DCED 4C                    DATA   :4C         L
DCEE 45                    DATA   :45         E
DCEF 58                    DATA   :58         X
DCF0 00                    DATA   :00
                    *
DCF1 DC03           ERMUA  DATA   :DC,:03     UNDEFINED
DCF3 20                    DATA   :20
DCF4 41                    DATA   :41         A
DCF5 52                    DATA   :52         R
DCF6 52                    DATA   :52         R
DCF7 41                    DATA   :41         A
DCF8 59                    DATA   :59         Y
DCF9 00                    DATA   :00
                    *
DCFA DD0A           ERML0  DATA   :DD,:0A     LOADING ERROR
DCFC 30                    DATA   :30         0
DCFD 00                    DATA   :00
                    *
DCFE DD0A           ERML1  DATA   :DD,:0A     LOADING ERROR
DD00 31                    DATA   :31         1
DD01 00                    DATA   :00
                    *
DD02 DD0A           ERML2  DATA   :DD,:0A     LOADING ERROR
DD04 32                    DATA   :32         2
DD05 00                    DATA   :00
                    *
DD06 DD0A           ERML3  DATA   :DD,:0A     LOADING ERROR
DD08 33                    DATA   :33         3
DD09 00                    DATA   :00
                    *
```

```
                     ERMLO
DD0A 8D1B            MSGL      DATA    :8D,:1B    LOAD
DD0C DBF3                      DATA    :DB,:F3    ING
DD0E DC15                      DATA    :DC,:15    ERROR
DD10 20                        DATA    :20
DD11 00                        DATA    :00
                     *
DD12 DC15            ERMEL     DATA    :DC,:15    ERROR
DD14 20                        DATA    :20
DD15 DB83                      DATA    :DB,:83    LINE
DD17 8BF2                      DATA    :8B,:F2    RUN
DD19 00                        DATA    :00
                     *
                     *
                     *
DD1A                           END

***************************
* S Y M B O L   T A B L E *
***************************

ERMBS  DC5C   ERMCN  DCA9   ERMDO  DC68   ERMEL  DD12
ERMID  DC79   ERMIN  DCB2   ERMLO  DCFA   ERML1  DCFE
ERML2  DD02   ERML3  DD06   ERMLN  DCD6   ERMLO  DD0A
ERMLS  DC9D   ERMNA  DCD8   ERMNC  DCC2   ERMNF  DC1C
ERMOD  DC33   ERMOF  DCB7   ERMOM  DC47   ERMOS  DC95
ERMOV  DC3E   ERMRG  DC2C   ERMSN  DC23   ERMSO  DC38
ERMTC  DCE3   ERMTM  DC8A   ERMUA  DCF1   ERMUS  DC50
MBREAK DBE0   MERROR DC15   MING   DBF3   MINVAL DC81
MLINE  DB83   MLINN  DC53   MLINR  DB98   MNUMBE DC55
MOUTOF DC0D   MSG01  DB6F   MSG02  DB7F   MSG03  DB89
MSG04  DB93   MSG05  DBA8   MSG06  DBB0   MSG07  DBB8
MSG09  DBC5   MSG10  DBC9   MSG11  DBCF   MSG14  DBDB
MSG15  DB9C   MSG17  DBC0   MSGL   DD0A   MSGOR  DCDB
MSTRIN DBF0   MTAPE  DBFD   MTYPE  DBF7   MUNDF  DC03
MWITHO DBE6   RMS01  DB6F
```

```
                            ORG    :DD1A
                    *
                    *
                    *
                    ******************
                    * INPUT TEXT LINE *
                    ******************
                    *
                    * Part of 'restart interpreter' (C853).
                    * Scans keyboard and reads in a line on the current
                    * screen line, up until car.ret. First prints
                    * car.ret and a prompt ('*').
                    * The routine can be aborted on car.ret or Break
                    * only.
                    *
                    * Entry: A: Contains prompt.
                    * Exit:  CY=1: Break pressed.
                    *        CY=0: HL:  Address 1st character on line.
                    *                C=1: Offset 1st significant characte
                    *                ABDEHL preserved.
                    *
DD1A F5             INPLO   PUSH   PSW          
DD1B CD55DD                 CALL   :DD55        Cursor to column 0
DD1E F1                     POP    PSW          Get prompt
DD1F 37             INPLN   STC                 CY=1
DD20 F5                     PUSH   PSW
DD21 E5                     PUSH   H            Save cursor coord 1st char
DD22 CD6ADD         IPL10   CALL   :DD6A        Print prompt
DD25 21B902                 LXI    H,:02B9
DD28 3600                   MVI    M,:00        Enable complete keyb.scan
DD2A CDBED6         IPL20   CALL   :D6BE        Get keyb. input
DD2D DA49DD                 JC     :DD49        Ignore line if Break presse
DD30 CA2ADD                 JZ     :DD2A        Wait for input
DD33 FE20                   CPI    :20          Printable character?
DD35 D222DD                 JNC    :DD22        Print it and get next one
DD38 FE08                   CPI    :08          Backspace
DD3A CA22DD                 JZ     :DD22        Print it; get next char
DD3D FE0D                   CPI    :0D          Car.ret?
DD3F C22ADD                 JNZ    :DD2A        Get next char if not

                    * Exit on car.ret:

DD42 35                     DCR    M            Set KBRFL for BREAK only
DD43 0E01                   MVI    C,:01
DD45 E1             EXIT1   POP    H            Get cursor coord 1st char
DD46 F1                     POP    PSW          Get prompt
DD47 3F                     CMC                 CY=0
DD48 C9                     RET

                    * Exit on Break:

DD49 35             IPL30   DCR    M            Set KBRFL for BREAK only
DD4A 3E21                   MVI    A,:21
DD4C CD60DD                 CALL   :DD60        Print '!'
DD4F CD5EDD                 CALL   :DD5E        Print car.ret
DD52 E1                     POP    H
DD53 F1                     POP    PSW          Get prompt, CY=1
DD54 C9                     RET
                    *
                    **********************
                    * CURSOR TO COLUMN 0 *
                    **********************
```

```
                        *
                        * The X-coordinate of the cursor is checked.
                        * If not 0, a car.ret is printed.
                        *
                        * Entry: None.
                        * Exit:  AF corrupted. BCDEHL preserved.
                        *
DD55 D5                 COLO    PUSH  D
DD56 E5                         PUSH  H
DD57 EF                         RST   5        Get cursor pos (HL) and size
DD58 0C                         DATA  :0C      char.screen (DE)
DD59 7D                         MOV   A,L      X-coord cursor in A
DD5A E1                         POP   H
DD5B D1                         POP   D
DD5C B7                         ORA   A
DD5D C8                         RZ             Abort if cursor in Col.0
DD5E 3E0D               CRLF    MVI   A,:0D    Else: print CR
                        *
                        * GENERAL OUTPUT ROUTINE:
                        *
                        * Outputs a character in a direction depending
                        * on OTSW (#0131).
                        *
                        * Entry: A: Character to be transmitted.
                        * Exit:  AF corrupted.
                        *
                        SCCHR
DD60 F5                 OUTC    PUSH  PSW      Preserve char
DD61 3A3101                     LDA   :0131
DD64 FE02                       CPI   :02      Check output direction
DD66 D270DD                     JNC   :DD70    If to edit buf/DOUTC

                        * To screen/RS232 - OTSW=0/1:

DD69 F1                         POP   PSW      Get char
DD6A EF                 COUTC   RST   5        Character to screen
DD6B 03                         DATA  :03
DD6C D442D6                     CNC   :D642    Output to RS232 if reqd
DD6F C9                         RET

                        * To DOUTC - OTSW=3:

DD70 00                 OTC10   NOP
DD71 C24CD7                     JNZ   :D74C    Character to DOUTC

                        * To editbuffer - OTSW=2:

DD74 F1                         POP   PSW      Get char
DD75 F5                 OTBIN   PUSH  PSW
DD76 E5                         PUSH  H
DD77 D5                         PUSH  D
DD78 2AA400                     LHLD  :00A4    Get edit input pointer
DD7B 77                         MOV   M,A      Byte in edit buffer
DD7C 23                         INX   H
DD7D 22A400                     SHLD  :00A4    Update edit input pointer
DD80 EB                         XCHG           in DE
DD81 2AA600                     LHLD  :00A6    Get end edit buffer
DD84 CD14DE                     CALL  :DE14    Calculate free buffer space
DD87 DA8EDD                     JC    :DD8E    If edit buffer full
DD8A D1                         POP   D
DD8B E1                         POP   H
                        OTC99   POP   PSW
```

```
DD8D C9                   RET
                  
                  * If edit buffer full:
                  
DD8E CDCADE       OTC20   CALL   :DECA      Heap back to right size
DD91 C310DA               JMP    :DA10      Run error 'OUT OF MEMORY'
                  *
                  *******************
                  * OUTPUT TO RS232 *
                  *******************
                  *
                  * Transmits a character to the RS232 interface via
                  * the TICC if the interface is ready for it.
                  * In case of a car.ret, also a line feed is send.
                  *
                  * Entry: A: Character to be transmitted.
                  * Exit:  ABCDEHL preserved, F corrupted.
                  *
DD94 F5           OUTSE   PUSH   PSW        Preserve char
DD95 3A00FD       OTS10   LDA    :FD00
DD98 E608                 ANI    :08        Check peripheral ready
DD9A CA95DD               JZ     :DD95      Wait untill ready
DD9D 3AF3FF       OTS20   LDA    :FFF3
DDA0 E610                 ANI    :10        Check TICC buffer empty
DDA2 CA9DDD               JZ     :DD9D      Wait untill empty
DDA5 F1                   POP    PSW        Get char
DDA6 32F6FF               STA    :FFF6      Load serial output buffer
DDA9 FE0D                 CPI    :0D        Carriage return?
DDAB C0                   RNZ               Ready if not
                  
                  * If car.ret:
                  
DDAC F5                   PUSH   PSW
DDAD 3E0A                 MVI    A,:0A
DDAF CD94DD               CALL   :DD94      Send line feed too
DDB2 F1                   POP    PSW
DDB3 C9                   RET
                  *
                  ********************
                  * INPUT FROM RS232 *
                  ********************
                  *
                  * Gets inputs from RS232 via TICC. Only 7-bit
                  * Ascii-code is accepted.
                  *
                  * Entry: No conditions.
                  * Exit:  A: Character received (0 if nothing).
                  *           BCDEHL preserved.
                  *
                  CINC
DDB4 3AF3FF       INSER   LDA    :FFF3
DDB7 E608                 ANI    :08        Check if something received
DDB9 C8                   RZ                Abort if no reception
DDBA 3AF0FF               LDA    :FFF0      Received char in A
DDBD E67F                 ANI    :7F        Mask bit 7
DDBF C9                   RET
                  *
                  ********************************
                  * RS232 FRAME ERROR - (not used) *
                  ********************************
                  *
                  * Break test for serial input line.
                  *
DDC0 3AF3FF       BRSER   LDA    :FFF3
DDC3 1F                   RAR
DDC4 D0                   RNC             Abort if no break
DDC5 3AF3FF       BRS10   LDA    :FFF3    If break: check again
DDC8 1F                   RAR
DDC9 DAC5DD               JC     :DDC5    Wait until end of break
DDCC 3AF0FF               LDA    :FFF0    Load received character
DDCF 3F                   CMC             Set CY=1
DDD0 C9                   RET
                  *
                  *
                  *  ===========================
                  *** ENCODING SERVICE ROUTINES ***
                  *  ===========================
                  *
                  *
                  * The following routines are used both in 'main'
                  * and 'decode' modules.
                  *
                  ************************************************
                  * GET CHARACTER FROM LINE, NEGLECT TAB + SPACE *
                  ************************************************
                  *
                  * Entry: C: Position on current line.
                  * Exit:  Character in A; tab and space neglected.
                  *        C: Points to next character.
                  *        BDEHL preserved.
                  *
DDD1 0C           IGNBR   INR    C        Pnts to next char
DDD2 CDE0DD       IGNB    CALL   :DDE0    Get char from line
DDD5 FE20                 CPI    :20      Space?
DDD7 CAD1DD               JZ     :DDD1    Then get next char
DDDA FE09                 CPI    :09      Tab?
DDDC CAD1DD               JZ     :DDD1    Then get next char
DDDF C9                   RET
                  *
                  ***************************
                  * GET CHARACTER TO ENCODE *
                  ***************************
                  *
                  * Returns a character from some position on the
                  * current line. The source is determined by EFSW.
                  *
                  * Entry: C: Position on current line (max. 219).
                  * Exit:  EFSW=0  - Keyboard: Char on line pos in A.
                  *        EFSW>=2 - Edit buf: Char on EFEPT + line
                  *                            pos in A.
                  *        EFSW=1  - String:   Idem. If COUNT=line pos
                  *                            then char is car.ret.
                  *        F corrupted, BCDEHL preserved.
                  *
DDE0 3A3501       EFETCH  LDA    :0135
DDE3 FE01                 CPI    :01      Check input direction
DDE5 DAFFDD               JC     :DDFF    If from keyb/RS232
DDE8 C2F4DD               JNZ    :DDF4    If from edit buffer

                  * If from string:

DDEB 3A3401               LDA    :0134    If string: Get COUNT
DDEE B9                   CMP    C        COUNT=pos.on curr.line?
DDEF 3E0D                 MVI    A,:0D    Then char is car.ret
```