```
0001  STM80
01                ;                                        RCSL: 43-GLXXXX
02                ;                                        AUTHOR: DHA
03                ;                                        EDITED: 77.04.20
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19                ;                        SXX04
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37                ; KEYWORDS: MUS, OPERATING SYSTEM, LISTING.
38                ; ABSTRACT: MUS OPERATING SYSTEM.
39                ; ASCII PAPER TAPE: RCSL 43-GLXXXX
40                ; REL. BINARY PAPER TAPE RCSL 43-GLXXXX
```

```
| 0002  STM80
01
02                ;    ******** SYSTEM ********
03
04          000000 .IFE SOPDEV*10+SINDEV-5
05                 .TITLE STM80
06                 .ENDC
07                 .IFE SOPDEV*10+SINDEV
08                 .TITL STP05
09                 .ENDC
10                 .IFE SOPDEV*10+SINDEV-1
11                 .TITL STM05
12                 .ENDC
13                 .IFE SOPDEV*10+SINDEV-2
14                 .TITL STC05
15                 .ENDC
16                 .IFE SOPDEV*10+SINDEV-3
17                 .TITL STF05
18                 .ENDC
19                 .IFE SOPDEV*10+SINDEV-4
20                 .TITLE STFA0
21                 .ENDC
22                 .IFE SOPDEV*10+SINDEV-10
23                 .TITL SOP05
24                 .ENDC
25                 .IFE SOPDEV*10+SINDEV-11
26                 .TITL SOM05
27                 .ENDC
28                 .IFE SOPDEV*10+SINDEV-13
29                 .TITL SOF05
30                 .ENDC
31
```

```
| 0003  STM80
01
02                .NREL
03          000001 .TXTM 1
04          000012 .RDX 10
05
06                S4:                              ; PROGRAM:
07  00000'100001         1B0+0B1+0B2+1              ;   SPECIFICATION
08  00001'001142'         S80                       ;   START
09  00002'000000          0                         ;   CHAIN
10  00003'001361          S0-S4                     ;   SIZE
11                .TXT .S<0><0><0><0>.              ;   NAME
    00004'051400
    00005'000000
    00006'000000
12
13                ; THE ASSEMBLY CONSTANTS SINDEV , SOPDEV AND AUTO SHOULD BE DEFIN
14                ; ON A SEPERATE TAPE
15                ;
16                ; SOPDEV = 0   OPERATOR DEVICE TTY
17                ;         = 1        -      -    OCP
18                ;
19                ; SINDEV = 0   INPUT    DEVICE PTR
20                ;         = 1     -            -    MT0
21                ;         = 2     -            -    CDR
22                ;         = 3     -            -    KIT+FD0
23                ;         = 4     -            -    FD0
24                ;         = 5     -            -    MT ( SPECIAL RC8000 FE. SYSTEM)
25                ;
26                ; AUTO    = 1  AUTOMATIC INT. ON FD0 AND MT0
27                ;         = 0  NOT AUTOMATIC INT. ON FD0 AND MT0
```

OPNAM = S or Q the process name

S = 0
Q = 1

~~OPNAM = "S - "S*OPNAM + "Q*OPNAM~~

OPNAM = "S + ("Q - "S)*OPNAM

```
.DO  OPNAM == S
  . DO  AUTO == 0
    .TITLE .SS006.
  .ENDC
  .DO
    .ENDC   .SQA005
.ENDC

.DO  OPNAM == Q
        .SQ005
        .SQA005
```

```
| 0004  STM80
01                S05:                          ; INZONE DESCRIPTOR:
02                .IFE SINDEV                   ;   NAME
03                .TXT .PTR<0><0>.
04                .ENDC
05                .IFE SINDEV-1
06                .TXT .MT0<0><0>.
07                .ENDC
08                .IFE SINDEV-2
09                .TXT .CDR<0><0>.
10                .ENDC
11                .IFE SINDEV-3
12                .TXT .KIT<0><0>.
13                .ENDC
14                .IFE SINDEV-4
15                .TXT .FD0<0><0>.
16                .ENDC
17        000000  .IFE SINDEV-5
18                .TXT .MT<0><0><0>.
   00007'046524
   00010'000000
   00011'000000
19                .ENDC
```

.TXT .<0><0><0><0><0>.

S05: .TXT '<0><0><0><0><0>.

```
I 0005   STM80
   00012'000032          S051-S05                     ;    SIZE
   00013'000001          1                            ;    MODE
         000002  .RDX 2
         000017  .IFN SINDEV=10*SINDEV
   00014'000016          D10                          ;    KIND
   00015'141733          1100001111011011             ;    MASK
                 .ENDC
                 .IFE SINDEV-10*SINDEV
                                            1
                         1110001111111110
                 .ENDC
         000012  .RDX 10
   00016'000150'         S059                         ;    GIVEUP
   00017'000000          0                            ;    FILE
   00020'000000          0                            ;    BLOCK
   00021'000000          0                            ;    CONVERSION
   00022'000041'         S051                         ;    BUFFER
   00023'000107          S059-S051                    ;    SIZE OF BUFFER
   00024'000000          0                            ;    FORMAT
   00025'000000          0                            ;    LENGTH
   00026'000120"         S052                         ;    FIRST
   00027'000120"         S052                         ;    TOP
   00030'000041'         S051                         ;    USED SHARE
   00031'000200          S050  0                      ;    SHARE LENGTH  <-
   00032'000000          0                            ;    REMAINING
                 S0592:
         000006  .BLK    ZAUX                         ;    AUXILIARY
```

%

```
1 0006  STM80
01
02
03                  S051:                       ; INPUT SHARE DESCRIPTOR:
04  00041'000000          0                     ;   OPERATION
05  00042'000000          0                     ;   COUNT
06  00043'000000          0                     ;   ADDRESS
07  00044'000000          0                     ;   SPECIAL
08  00045'000041'         S051                  ;   NEXT SHARE
09  00046'000000          0                     ;   STATE
10  00047'000120"         S052                  ;   FIRST SHARED
11
12                  .IFE SINDEV/3               ;   IF DEVICE=PTR,MT,CDR THEN
13                  S050=80                     ;     SHARE LENGTH:= 80;
14                  .ENDC
15          000000  .IFE SINDEV/3-1             ;   IF DEVICE=FD0,DP0 THEN
16          000200  S050=128                    ;     SHARE LENGTH:=128;
17                  .ENDC
18
19          000120"                             ; FIRST SHARED:
20          000100  .BLK  S050*1/2              ;   MAKE ROOM FOR SHARE;
21
```

S052 = 55*2

```
| 0007  STM80
01                S059:                              ; GIVEUP ACTION:
02  00150'054462         STA     3    $0591         ;   SAVE LINK;
03  00151'141120         MOVZL   2,0                ;
04  00152'006465         JSR@         $0595         ;   FIRSTMESSAGE(ZONE.ZNAME*2);
05  00153'020471         LDA     0    $0598         ;
06  00154'006464         JSR@         $0595+1       ;   CONTMESSAGE('ERROR ');
07  00155'024656         LDA     1    $0592         ;   RESTORE(STATUS);
08  00156'102000         ADC     0,0                ;   BIT:=-1;
09  00157'101440         INCO    0,0                ;   REPEAT BIT:=BIT+1;
10  00160'125103         MOVL    1,1  SNC           ;   STATUS:= STATUS SHIFT 1
11  00161'000776         JMP          .-2           ;   UNTIL CARRY;
12  00162'024466         LDA     1   .50            ;   FIRSTCHAR:=50;
13  00163'034126         LDA     3   .10             ;
14  00164'162423         SUBZ    3,0  SNC           ;   IF BIT>10 THEN
15  00165'163001         ADD     3,0  SKP           ;    BEGIN
16  00166'125400         INC     1,1                ;    BIT:=BIT-10; FIRSTCHAR:=FIRSTCHAR
17  00167'006212         OUTCHAR                    ;    END;
18  00170'024135         LDA     1   .48            ;
19  00171'107000         ADD     0,1                ;   OUTCHAR(OPZONE,FIRSTCHAR);
20  00172'006212         OUTCHAR                    ;   OUTCHAR(OPZONE,BIT+48);
21  00173'006446         JSR@         $0595+2       ;   ENDMESSAGE;
22  00174'006205         INBLOCK                    ;   INBLOCK(OPZONE);
23  00175'102400         SUB     0,0                ;
24  00176'041023         STA     0    ZREM,2        ;   REM.OPZONE:=0;
25  00177'020513         LDA     0    $0593-1       ;
26  00200'024434         LDA     1    $0594         ;
27  00201'106404         SUB     0,1  SZR           ;   IF ZONE.BUF(0:1)<>'ST' THEN
28  00202'000410         JMP          $102          ;    GOTO WRONG;
29  00203'020510         LDA     0    $0593         ;
30  00204'024431         LDA     1    $0594+1       ;
31  00205'106405         SUB     0,1  SNR           ;   IF ZONE.BUF(2:3)='AR' THEN
32  00206'000410         JMP          $104          ;    GOTO REP;
33  00207'024427         LDA     1    $0594+2       ;
34  00210'106405         SUB     0,1  SNR           ;   IF ZONE.BUF(2:3)='OP' THEN
35  00211'002431         JMP@         $0596         ;    GOTO NEXT COMMAND;
36  00212'020421  $102:  LDA     0    $0599         ; WRONG:
37  00213'006424         JSR@         $0595         ;   FIRSTMESS(SYNTAX);
38  00214'006425         JSR@         $0595+2       ;   ENDMESS;
39  00215'002425         JMP@         $0596         ;   GOTO NEXT COMMAND;
40  00216'030425  $104:  LDA     2    $0597         ; REP: ZONE:=IN;
41  00217'021024         LDA     0    Z0,2          ;
42  00220'024107         LDA     1    $ILLEGAL      ;
43  00221'107405         AND     0,1  SNR           ;   IF STATUS AND ILLEGAL THEN
44  00222'000405         JMP          .+5           ;    BEGIN
45  00223'024054         LDA     1    PROCESS       ;
46  00224'006010         SEARCHITEM                 ;   SEARCHITEM(ZONE,PROCESSCHAIN);
47  00225'151004         MOV     2,2  SZR           ;    IF PROC<>0 THEN
48  00226'006011         CLEANPROCESS               ;   CLEANPROCESS(PROC);
49  00227'030414         LDA     2    $0597         ;    END;
50  00230'034402         LDA     3    $0591         ;
51  00231'002203  $105:  .REPEATSHARE               ;   REPEATSHARE(ZONE);
52
```

```
| 0008  STM80
01
02 00232'000000 S0591:0           ; LINK
03 00233'002554"S0599: S900*2+2    ;
04 00234'051524 S0594:"S*256+"T   ;
05 00235'040522        "A*256+"R   ;
06 00236'047520        "O*256+"P   ;
07 00237'001333:S0595:S95          ; FIRSTMESSAGE
08 00240'001345'       S96
09 00241'001350'       S97
10 00242'001142'S0596:S80         ; REF.NEXT COMMAND
11 00243'000007:S0597:S05         ; IN ZONE
12 00244'000512"S0598:.+1*2        ; 'ERROR '
13                .TXT   .ERROR .
   00245'042522
   00246'051117
   00247'051040
   00250'000000
14        000250'.LOC  .=1
15 00250'000062 .50:   50
```

```
                         S08:                          ; OPERATOR ZONE DESCRIPTOR
                                                       ; NAME
             000000 .IFE SOPDEV
                    .TXT  .TTY<0><0>.
  00251'052124
  00252'054400
  00253'000000
                    .ENDC
                    .IFE SOPDEV=1
                    .TXT  .OCP<0><0>.


                    .ENDC
  00254'000032          S081-S08                       ; SIZE
  00255'000001          1                              ; MODE
  00256'000001          1                              ; KIND
  00257'000000          0  1                           ; MASK
  00260'000000          0  S105                        ; GIVEUP
  00261'000000          0                              ; FILE
  00262'000000          0                              ; BLOCK
  00263'000000          0                              ; CONVERSION
  00264'000303'         S081                           ; BUFFER
  00265'000017          S089-S081                      ; SIZE OF BUFFER
  00266'000000          0                              ; FORMAT
  00267'000000          0                              ; LENGTH
  00270'000624"         S082                           ; FIRST
  00271'000624"         S082                           ; TOP
  00272'000303'         S081                           ; USED SHARE
  00273'000020          S080                           ; SHARELENGTH
  00274'000000          0                              ; REMAINING
             000006 .BLK  ZAUX                         ;

                    S081:                              ; SHARE DESCRIPTOR
             000004 .BLK 4                             ;
  00307'000303'         S081                           ; NEXT SHARE
  00310'000000          0                              ; STATE
  00311'000624"         S082                           ; FIRST SHARED

             000020 S080=16                            ; SHARE LENGTH=16

             000624"S082=.*2                           ; FIRST SHARED
                    .TXT .INTLSSYSI[illegible]
  00312'044516
  00313'052040                   OPNAM = 256+"S
  00314'051523                   .TXT .YSI<137>.
  00315'054523
  00316'044415
  00317'000000
             000002 .BLK  S080+1/2-6
             000313'S0593=S082/2+1
                    S089:
```

.TXT .<0><0><0><0><0>.

```
| 0010  STM80
01                S06:
02                .IFE SINDEV
03                .TXT  .PTR<0><0>.          ; PRIMARY INPUT:
04                .ENDC
05                .IFE SINDEV-1
06                .TXT  .MT0<0><0>.
07                .ENDC
08                .IFE SINDEV-2
09                .TXT  .CDR<0><0>.
10                .ENDC
11                .IFE SINDEV-3
12                .TXT  .KIT<0><0>.
13                .ENDC
14                .IFE SINDEV-4
15                .TXT  .FD0<0><0>.
16                .ENDC
17        000000  .IFE SINDEV-5
18                .TXT .MT<0><0><0>.
   00322'046524
   00323'000000
   00324'000000
19                .ENDC
```

.TXT.<0><0><0><0><0>.

```
| 0011  STM80
01                    SCOMMANDS:                    ; COMMANDS:
02 00325'000330'            .+3
03 00326'000614'             S20
04                    .TXT .LOAD<0>.                ;   LOAD
   00327'046117
   00330'040504
   00331'000000
05 00332'000335'            .+3
06 00333'000713'             S50
07                    .TXT .KILL<0>.                ;   KILL
   00334'045511
   00335'046114
   00336'000000
08 00337'000342'             .+3
09 00340'000756'             S60
10                    .TXT .LIST<0>.                ;   LIST
   00341'046111
   00342'051524
   00343'000000
11 00344'000347'             .+3
12 00345'001122'             S75
13                    .TXT .INT<0><0>.              ;   INT
   00346'044516
   00347'052000
   00350'000000
14 00351'000354'             .+3
15 00352'000661'             S30
16                    .TXT .START.                  ;   START
   00353'051524
   00354'040522
   00355'052000
17 00356'000361'             .+3
18 00357'000665'             S40
19                    .TXT .STOP<0>.                ;   STOP
   00360'051524
   00361'047520
   00362'000000
20 00363'000366'             .+3
21 00364'000751'             S55
22                    .TXT .BREAK.                  ;   BREAK
   00365'041122
   00366'042501
   00367'045400
23 00370'000373'             .+3
24 00371'001012'             S70
25                    .TXT .IN<0><0><0>.            ;   IN
   00372'044516
   00373'000000
   00374'000000
```

```
| 0012  STM80
01 00375'000400'        .+3
02 00376'000672'        S500
03                .TXT .CLEAR.                  ;   CLEAR
   00377'041514
   00400'042501
   00401'051000
04 00402'000405'        .+3
05 00403'001130'        S755
06                .TXT .BEGIN.                  ;   BEGIN
   00404'041105
   00405'043511
   00406'047000
07 00407'000000         0
08 00410'001151'        S805
09                .TXT .END<0><0>.              ;   END
   00411'042516
   00412'042000
   00413'000000
10
```

```
| 0013  STM80
01                ;       ****** RELOCATABLE LOADER ******
02
03 00414'030470 S21:  LDA     2   SINZONE    ;
04 00415'102400       SUB     0,0             ; BLOCK:
05 00416'040476       STA     0   SCHECKSUM   ;   CHECKSUM:= 0;
06                                            ; LEADER:
07 00417'006207 S210: INCHAR                  ;   INCHAR(INZONE,CHAR);
08 00420'125005       MOV     1,1 SNR         ;   IF CHAR=0 THEN
09 00421'000776       JMP         S210        ;     GOTO LEADER;
10 00422'004534       JSR         S130        ;   COMPLETE WORD(CHAR,WORD);
11 00423'044421       STA     1   SBL         ;   BLOCK:= WORD;
12 00424'004534       JSR         S13         ;   GETWORD(WORD);
13 00425'044466       STA     1   SWORDCOUNT;   WORDCOUNT:= WORD;
14 00426'004532       JSR         S13         ;   GETWORD(WORD);
15 00427'044461       STA     1   SFLA0       ;   0.FLAGS:= WORD;
16 00430'004530       JSR         S13         ;   GETWORD(WORD);
17 00431'044460       STA     1   SFLA1       ;   1.FLAGS:= WORD;
18 00432'004526       JSR         S13         ;   GETWORD(WORD);
19 00433'044457       STA     1   SFLA2       ;   2.FLAGS:= WORD;
20 00434'004524       JSR         S13         ;   GETWORD(WORD);
21 00435'024452       LDA     1  .SFLAGS      ;   COMMENT: CHECKSUM NOT SAVED;
22 00436'044450       STA     1   SFLAGS      ;   FLAGS ADDR:= REF.FLAGS;
23 00437'024405       LDA     1   SBL         ;
24 00440'004411       JSR         S211        ;   INDEX:= TABLE;
25 00441'000002       2
26 00442'000006       6
27 00443'000007       7
28 00444'000000 SBL:  0
29 00445'000456'      S22   ; DATA
30 00446'000521'      S26   ; START
31 00447'000552'      S27   ; SKIP TITLE
32 00450'001246'.S907:S907  ; ILLEGAL
33 00451'021400 S211: LDA     0  +0,3         ; NEXT:
34 00452'122415       IEQ     1,0             ;   IF BLOCK=INDEX(0) THEN
35 00453'003404       JMP@       +4,3         ;     GOTO INDEX(4);
36 00454'175400       INC     3,3             ;   INDEX:= INDEX+1;
37 00455'000774       JMP         S211        ;   GOTO NEXT;
```

```
                 S22:                               ; DATA:
   00456'004502         JSR        S13         ;   GETWORD(WORD);
   00457'004514         JSR        S14         ;   ADJUST(WORD);
   00460'044435         STA    1   SCOREADDR   ;   COREADDR:= WORD;
   00461'030432         LDA    2   SWORDCOUNT  ;
   00462'146400         SUB    2,1             ;   MAX:= COREADDR-WORDCOUNT;
   00463'020422         LDA    0   SNMAX       ;   IF NMAX<MAX THEN
   00464'122433         SUBZ#  1,0 SNC         ;   NMAX:=MAX;
   00465'121000         MOV    1,0             ;
   00466'040417         STA    0   SNMAX       ;
   00467'036431         LDA@   3   .S04        ;
   00470'162432         SUBZ#  3,0 SZC         ;   IF NMAX>=PARAMTOP THEN
   00471'002425         JMP@       .S903       ;   GOTO REF.SIZE;
   00472'000405         JMP        S221        ;   GOTO TEST WORD COUNT;
                 S220:                         ; NEXT WORD:
   00473'004465         JSR        S13         ;   GETWORD(WORD);
   00474'004477         JSR        S14         ;   ADJUST(WORD);
   00475'046420         STA@   1   SCOREADDR   ;   0.COREADDR:= WORD;
   00476'010417         ISZ        SCOREADDR   ;   INCR(COREADDR);
   00477'030405 S221:   LDA    2   SINZONE     ; TEST WORD COUNT:
   00500'010413         ISZ        SWORDCOUNT  ;   IF INCR(WORDCOUNT)<>0 THEN
   00501'000772         JMP        S220        ;     GOTO NEXT WORD;
   00502'126000         ADC    1,1             ;   WORD:= -1;
   00503'000420         JMP        S260        ;   GOTO CHECK;

   00504'000007'SINZO:S05                      ; INZONE

         000057 SNSTART=EFIRST                 ; NSTART
   00505'000000 SNMAX:0                        ; NMAX

   00506'000000 SFLAG:0                        ; FLAGS ADDR
   00507'000510'.SFLA:SFLA0                    ; REF.FLAGS
   00510'000000 SFLA0:0                        ; 0.FLAGS
   00511'000000 SFLA1:0                        ; 1.FLAGS
   00512'000000 SFLA2:0                        ; 2.FLAGS
   00513'000000 SWORD:0                        ; WORDCOUNT
   00514'000000 SCHEC:0                        ; CHECKSUM
   00515'000000 SCORE:0                        ; COREADDR

   00516'001254'.S903:S903                     ; REF.SIZE
   00517'001251'.S904:S904                     ; REF.CHECKSUM
   00520'001242'.S04: S04
```

```
| 0015  STM80
01                S26:                              ; START:
02  00521'004437        JSR         S13             ;   GETWORD(WORD);
03  00522'004451        JSR         S14             ;   ADJUST(WORD);
04  00523'020771 S260:  LDA     0   SCHECKSUM        ; CHECK:
05  00524'101004        MOV     0,0 SZR             ;   IF CHECKSUM<>0 THEN
06  00525'002772        JMP@        .S904           ;     GOTO CHECKSUM;
07  00526'124015        COM#    1,1 SNR             ;   IF WORD=-1 THEN
08  00527'000665        JMP         S21             ;     GOTO BLOCK;
09
10  00530'125135        MOVZL#  1,1 SNR             ;   IF PROC=1B0 OR 0 THEN
11  00531'000416        JMP          S262           ;     GOTO UPDATE;
12  00532'131000        MOV     1,2                 ;
13  00533'050757 S261:  STA     2   SFLA2           ;   CHAIN.OLD:= PROC;
14  00534'020753        LDA     0  .SFLA0           ;   OLD:= REF.FLAG0;
15  00535'024054        LDA     1   PROCESS         ;   CHAIN:= PROCESS CHAIN;
16  00536'006015        RECHAIN                     ;   RECHAIN(OLD,CHAIN,PROC);
17  00537'020101        LDA     0  .1B0             ;
18  00540'041013        STA     0   STATE,2         ;   STATE.PROC:=STOPPED;
19  00541'006014        STARTPROCESS                ;   STARTPROCESS(PROC);
20  00542'031012        LDA     2   PROG,2          ;   PROG:=PROG.PROC;
21  00543'050747        STA     2   SFLA2           ;   CHAIN.OLD:=PROG;
22  00544'020743        LDA     0   .SFLA0         ;   OLD:=REF.SFLA0
23  00545'024071        LDA     1   PROGRAM         ;   NEW:= PROGRAM CHAIN;
24  00546'006015        RECHAIN                     ;   RECHAIN(OLD,NEW,PROG);
25  00547'020736 S262:  LDA     0   SNMAX           ; UPDATE:
26  00550'040057        STA     0   SNSTART        ;   NSTART:= NMAX;
27  00551'000460        JMP         S200            ;   GOTO LOAD NEXT;
28
29  00552'004406 S27:   JSR         S13             ; SKIP:
30  00553'010740        ISZ         SWORDCOUNT      ;   GETWORD(WORD);
31  00554'000776        JMP         S27             ;   IF INCR(COUNT)<>0 THEN GOTO SKIP;
32  00555'000637 .S21:  JMP         S21             ;   GOTO BLOCK;
```

```
| 0016  STM80
01                ; PROCEDURE GETWORD(WORD);
02                ; PROCEDURE COMPLETE WORD(CHAR,WORD);
03                ;         CALL:                 RETURN:
04                ; AC0                           DESTROYED
05                ; AC1     CHAR                  WORD
06                ; AC2     ZONE                  ZONE
07                ; AC3     LINK                  DESTROYED
08
09 00556'055031 S130: STA     3    Z5,2       ;
10 00557'000403       JMP          S131
11
12 00560'055031 S13:  STA     3    Z5,2       ; GET WORD:
13 00561'006207       INCHAR                   ;   INCHAR(ZONE,CHAR);
14                S131:                        ; COMPLETE WORD:
15 00562'121300       MOVS    1,0              ;   WORD:= CHAR;
16 00563'041030       STA     0    Z4,2        ;
17 00564'006207       INCHAR                   ;   INCHAR(ZONE,CHAR);
18 00565'021030       LDA     0    Z4,2        ;
19 00566'107300       ADDS    0,1              ;   WORD:= CHAR SHIFT 8 + WORD;
20 00567'020725       LDA     0    SCHECKSUM   ;
21 00570'123000       ADD     1,0              ;   CHECKSUM:= CHECKSUM+WORD;
22 00571'040723       STA     0    SCHECKSUM   ;
23 00572'003031       JMP@         Z5,2        ;   RETURN;
24
25                ; PROCEDURE ADJUST(WORD);
26                ; GETS THE NEXT THREE FLAG BITS FROM THE FLAGS WORDS, UPDATES
27                ; FLAGS ADDR IF THE WORD WAS EMPTIED.
28
29 00573'020103 S14:  LDA     0   .182         ; GET FLAGS:
30 00574'032712       LDA@    2    SFLAGS      ;   WORD:= 0,FLAGS ADDR;
31 00575'151120 S140: MOVZL   2,2              ;   FLAGS:= WORD(0:2);
32 00576'101103       MOVL    0,0 SNC          ;   WORD:= WORD SHIFT 3;
33 00577'000776       JMP          S140        ;
34 00600'052706       STA@    2    SFLAGS      ;   0,FLAGS ADDR:= WORD;
35 00601'151005       MOV     2,2 SNR          ;   IF WORD=0 THEN
36 00602'010704       ISZ          SFLAGS      ;     INCR(FLAGS ADDR);
37
38                ; ADJUSTS THE WORD ACCORDING TO THE FLAG BITS.
39                ;         CALL:                 RETURN:
40                ; AC0                           DESTROYED
41                ; AC1     WORD                  WORD (ADJUSTED)
42                ; AC2                           DESTROYED
43                ; AC3     LINK                  DESTROYED
44
45                S150:                        ; ADJUST:
46 00603'101225       MOVZR   0,0 SNR          ;   IF FLAGS(13:14)=0 THEN
47 00604'001400       JMP        +0,3          ;     RETURN;
48 00605'030057       LDA     2    SNSTA       ;   BASE:= NSTART;
49 00606'101002       MOV     0,0 SZC          ;   IF FLAGS(15:15)=1 THEN
50 00607'151120       MOVZL   2,2              ;     BASE:= BASE*2;
51 00610'147000       ADD     2,1              ;   WORD:= BASE+WORD;
52 00611'101224       MOVZR   0,0 SZR          ;   IF FLAGS(13) THEN
53 00612'002636       JMP@       .S907         ;    GOTO ILLEGAL;
54 00613'001400       JMP        +0,3          ; RETURN;
55
56                ;     ****** END OF RELOCATABLE LOADER ******
```

```
| 0017  STM80
01
02 00614'102400 S20:  SUB     0,0             ; LOAD:
03 00615'040670       STA     0    SNMAX      ;   MAX:=0;
04 00616'006471       JSR@         .S11       ;
05 00617'006467       JSR@         .S113      ;   SETZONE;
06 00620'000735       JMP          .S21       ;   IF CHECKEMPTY THEN LOAD;
07 00621'004436       JSR          S208       ;   INIT(PREV,CUR);
08
09 00622'006466 S201: JSR@        .S101       ;   REPEAT
10 00623'000403       JMP          .+3        ;   IF CHECKPROC(CUR) THEN
11 00624'004421       JSR          S206       ;    CHAINOUT
12 00625'000402       JMP          .+2        ;   ELSE
13 00626'004424       JSR          S207       ;    NEXTELEMENT
14 00627'151004       MOV     2,2  SZR        ;   UNTIL CUR=0;
15 00630'000772       JMP          S201       ;
16
17 00631'004426 S200: JSR          S208       ; REP: INIT(PREV,CUR);
18 00632'151005       MOV     2,2  SNR        ;   IF CUR=0 THEN
19 00633'000444       JMP         .S80        ;     GOTO NEXT COMMAND;
20
21 00634'006456       JSR@        .S16        ;   SETIDENT;
22
23 00635'006454 S2001:JSR@        .S115       ;   IF CHECKIDENT THEN
24 00636'000403       JMP          S203       ;    BEGIN
25 00637'004406       JSR          S206       ;    CHAINOUT;
26 00640'000715       JMP          .S21       ;   LOAD; GOTO REP
27 00641'004411 S203: JSR          S207       ;    END; NEXTELEM;
28 00642'151004       MOV     2,2  SZR        ;   IF CUR<>0 THEN
29 00643'000772       JMP          S2001      ;    GOTO CHECKELEM;
30 00644'000765       JMP          S200       ;   GOTO REP;
31
32 00645'030555 S206: LDA     2    S030       ; CHAINOUT:
33 00646'025003       LDA     1    +3,2       ;   NEXT:=CHAIN.CUR;
34 00647'030641       LDA     2    SFLA0      ;
35 00650'045003       STA     1    +3,2       ;   CHAIN.PREV:=NEXT;
36 00651'000403       JMP          .+3        ;   CUR:=PREV;
37 00652'030550 S207: LDA     2    S030       ; NEXTELEM:
38 00653'050635 S2071:STA     2    SFLA0      ;   PREV:=CUR;
39 00654'031003       LDA     2    +3,2       ; SET CUR:
40 00655'050545       STA     2    S030       ;   CUR:=CHAIN.CUR;
41 00656'001400       JMP          0,3        ; RETURN;
42
43 00657'030542 S208: LDA     2    S03        ; INIT(CUR,PREV):
44                                            ;   PREV:=CUR:=COMMAND;
45 00660'000773       JMP          S2071      ;   GOTO SET CUR;
46
47 00661'006427 S30:  JSR@         .S101      ; START:
48 00662'002423       JMP@         .S902      ;   GET PROCESS(PROC,UNKNOWN);
49 00663'006014       STARTPROCESS            ;   STARTPROCESS(PROC);
50 00664'000413       JMP         .S80        ;   GOTO NEXT COMMAND;
51
52 00665'006423 S40:  JSR@         .S101      ; STOP:
53 00666'002417       JMP@         .S902      ;   GETPROCESS(PROC,UNKNOWN);
54 00667'004412       JSR          S502       ;    CHECKALLOWED(PROC);
55 00670'006013       STOPPROCESS             ;   STOPPROCESS(PROC);
56 00671'000406       JMP          .S80       ;   GOTO NEXT COMMAND;
```

```
| 0018  STM80
01                S500:                          ; CLEAR:
02  00672'020531          LDA     0    .SLAST     ;
03  00673'040057          STA     0    SNSTART    ;   NSTART:=LAST;
04  00674'030042          LDA     2    CUR+CHAIN
05  00675'034040          LDA     3    CUR        ;   PROC:=PROCESS.CHAIN;
06  00676'156405          SUB     2,3  SNR        ;   IF PROC=CUR THEN
07  00677'000520 .S80:    JMP        ..S80        ;   GOTO NEXT COMMAND;
08  00700'000420          JMP        S501         ;   GOTO KILL PROC;
09
10  00701'024040 S502:    LDA     1    CUR         ; CHECKALLOWED(PROC):
11  00702'146503          SUBL    2,1  SNC         ;   IF PROC<=CUR THEN
12  00703'002402          JMP@       .S902        ;     GOTO UNKNOWN;
13  00704'001400          JMP        0,3          ; RETURN;
14
15  00705'001260'.S902:   S902
16  00706'001077'.S113:   S113
17  00707'001045'.S11:    S11
18  00710'001105'.S101:   S101
19  00711'001103'.S115:   S115
20  00712'001066'.S16:    S16
```

```
| 0019  STM80
01                S50:                                ; KILL:
02  00713'006775        JSR@       .S101            ;   GET PROCESS(PROC,
03  00714'002771        JMP@       .S902            ;     +0: UNKNOWN);
04  00715'004764        JSR         S502            ;   CHECKALLOWED(PROC);
05  00716'036503        LDA@   3    S03             ;   COMMAND:=NEXT.COMMAND;
06  00717'054502        STA    3    S03             ;
07  00720'006013 S501:  STOPPROCESS                 ;   STOPPROCESS(PROC);
08  00721'006011        CLEANPROCESS                ;   CLEANPROCESS(PROC)
09  00722'024046        LDA    1    TOPTABLE        ;
10  00723'034045        LDA    3    TABLE           ;   INDEX:=DEVICE TABLE+3;
11  00724'021403 S509:  LDA    0   +3,3             ; NEXT:
12  00725'101220        MOVZR  0,0                  ;   ENTRY:=DEVTA(INDEX)//2;
13  00726'142404        SUB    2,0  SZR             ;   IF ENTRY = PROC THEN
14  00727'000403        JMP         .+3             ;    DEVTA(INDEX):=CUR*2;
15  00730'020112        LDA    0    CUR2            ;
16  00731'041403        STA    0     +3,3           ;
17  00732'175400        INC    3,3                  ;   INDEX:=INDEX+1;
18  00733'136414        SUB#   1,3  SZR             ;   IF INDEX<>TOPTABLE THEN
19  00734'000770        JMP         S509            ;   GOTO NEXT;
20  00735'102400        SUB    0,0                  ;
21  00736'040506        STA    0    SWORK+2         ;   CHAIN.NEW:=0;
22  00737'024500        LDA    1   .SWORK           ;   NEW:= IRR CHAIN;
23  00740'020054        LDA    0    PROCESS         ;   OLD:= PROCESS CHAIN;
24  00741'006015        RECHAIN                     ;   RECHAIN(OLD,NEW,PROC);
25  00742'031012        LDA    2    PROG,2          ;   PROG:=PROGRAM.CUR;
26  00743'020071        LDA    0    PROGRAM         ;   OLD:=PROGRAM CHAIN;
27  00744'034052        LDA    3    PFIRST
28  00745'156405        SUB    2,3  SNR            ;   IF PROG=OLD.CHAIN THEN
29  00746'050057        STA    2    SNSTART         ;   SNSTART:=PROG;
30  00747'006015        RECHAIN                     ;   RECHAIN(OLD,NEW,PROG);
31  00750'000567        JMP         .S801           ;   GOTO NEXTCOM1;
32
33                S55:                                ; BREAK:
34  00751'004534        JSR         S101            ;   GET PROCESS(PROC,
35  00752'002733        JMP@       .S902            ;     +0: UNKNOWN);
36  00753'020117        LDA    0   .2                ;
37  00754'006012        BREAKPROCESS                ;   BREAK PROCESS(PROC,2);
38  00755'000442        JMP        ..S80            ;   GOTO NEXT COMMAND;
```

```
I 0020  STM80
01                  S60:                            ; LIST:
02  00756'102400            SUB     0,0             ;
03  00757'006452            JSR@        .S95        ;   FIRSTMESS(0);
04  00760'034054            LDA     3   PROCESS      ;
05  00761'054460            STA     3   SLINK        ;   PROC:= PROCESS CHAIN;
06  00762'034457 S600:      LDA     3   SLINK        ; NEXT:
07  00763'035402            LDA     3   CHAIN,3      ;   PROC:= CHAIN.PROC;
08  00764'054455            STA     3   SLINK        ;
09  00765'175005            MOV     3,3 SNR          ;   IF PROC=0 THEN
10  00766'000416            JMP         S601         ;     GOTO OUT MAX;
11  00767'020116            LDA     0   .NAME        ;
12  00770'163120            ADDZL   3,0              ;   TEXT:= NAME.PROC;
13  00771'006215            OUTTEXT                  ;   OUTTEXT(OUTZONE,TEXT);
14  00772'006211            OUTSPACE                 ;   OUTSPACE(OUTZONE);
15  00773'020446            LDA     0   SLINK        ;
16  00774'006216            OUTOCTAL                 ;   OUTOCTAL(PROC);
17  00775'034444            LDA     3   SLINK        ;
18  00776'021430            LDA     0   RESER,3      ;
19  00777'024112            LDA     1   .64          ;
20  01000'101005            MOV     0,0 SNR          ;   IF RESERVER.PROC=0 THEN
21  01001'006212            OUTCHAR                  ;     OUTCHAR(OUTZONE,'@');
22  01002'006213            OUTNL                    ;   OUTNL(OUTZONE);
23  01003'000757            JMP         S600         ;   GOTO NEXT;
24                  S601:                           ; OUT MAX:
25  01004'020057            LDA     0   SNSTART      ;
26  01005'101120            MOVZL   0,0              ;   WORD:=MAX*2;
27  01006'006424            JSR@        .S93         ;   TEXT:=
28                         .TXT                              .MAX:.;  GOTO OUTERROR;
    01007'046501
    01010'054072
    01011'000000
29
30                  S70:                            ; IN:
31  01012'004473            JSR         S101         ;   GETPROCESS(PROC,UNKNOWN);
32  01013'002672            JMP#        S900 .+1     ;
33  01014'004501            JSR         S17          ;   CLOSECURZONE;
34  01015'024405            LDA     1   S030         ;
35  01016'004437            JSR         S110         ;   MOVENAME(CURRENT,INZONE);
36  01017'000523 ..S80:     JMP         S80          ;   GOTO NEXT COMMAND;
37
38                  ; VARIABLES AND CONSTANTS:
39
40  01020'001021'.S03:      .+1         ; REF.CURRENT COMMAND
41  01021'000000 S03:       0           ; CURRENT COMMAND
42  01022'000000 S030:      0           ; CURRENT PARAM
43
44  01023'001420'.SLAST:    SLAST
45  01024'001151'SENDC:     S805
46  01025'000007'.S05:      S05         ; ADDR.IN
47  01026'000322'.S06:      S06         ; ADDR.INPUT DESCRIPTION
48  01027'000000 .S07:      0           ; ADDR.CURRENT ZONE
49  01030'000251'.S08:      S08         ; ADDR.PRIMARY INPUT
50  01031'001333'.S95:      S95
51  01032'001274'.S93:      S93
```

```
| 0021  STM80
01 01033'001032'.SCOMM: .+1=CHAIN;
02 01034'000323'        SCOMMANDS=CHAIN
03
04 01035'001034'.SITEM: .+1=CHAIN
05 01036'001036'        .       ;
06 01037'001042'.SWORK: SWORK
07 01040'000000 SCHAIN: 0       ;
08 01041'000000 SLINK:  0       ;
09       000003 SWORK:  .BLK 3  ;
10
11                ; PROCEDURE SETZONE;
12                ;       CALL            RETURN
13                ; AC0                   DESTROYED
14                ; AC1                   DESTROYED
15                ; AC2                   ZONE
16                ; AC3   LINK            DESTROYED
17                ;
18 01045'054774 S11:    STA     3       SLINK   ; SETZONE:
19 01046'030757         LDA     2       .S05    ;
20 01047'050760         STA     2       .S07    ; USED ZONE:=INZONE;
21 01050'102520         SUBZL   0,0             ;
22 01051'006221         OPEN                    ; OPEN(ZONE,1);
23 01052'102520         SUBZL   0,0             ;
24 01053'041010         STA     0       ZFILE,2 ; FILE.ZONE:=1;
25 01054'002765         JMP@            SLINK   ; RETURN;
26
27 01055'054764 S110:   STA     3       SLINK   ; MOVENAME:
28 01056'135000         MOV     1,3             ;
29 01057'021400         LDA     0       +0,3    ;
30 01060'041000         STA     0       +0,2    ;
31 01061'021401         LDA     0       +1,3    ;
32 01062'041001         STA     0       +1,2    ; NAME.ZONE:=NAME.DESC ADDR;
33 01063'021402         LDA     0       +2,3    ;
34 01064'041002         STA     0       +2,2    ;
35 01065'002754         JMP@            SLINK   ; RETURN
```

```
I 0022  STM80
01                ; PROCEDURE SET IDENT;
02 01066'054753 S16:    STA     3       SLINK    ; SET IDENT;
03 01067'030740         LDA     2       .S07     ;
04 01070'025010         LDA     1       ZFILE,2  ;
05 01071'125400         INC     1,1              ;
06 01072'102520         SUBZL   0,0              ;
07 01073'006217         SETPOSITION              ; SETPOSITION(ZONE,FILE.ZONE+1,1)
08 01074'024743         LDA     1       .SWORK   ;
09 01075'006223         INNAME                   ; INNAME(ZONE,WORK);
10 01076'002743         JMP@            SLINK    ;  RETURN;
11
12 01077'030723 S113:   LDA     2       S030     ; CHECKEMPTY;
13 01100'151005         MOV     2,2     SNR      ; IF CUR PARAM=0 THEN
14 01101'001400         JMP             +0,3     ; RETURN;
15 01102'001401         JMP             +1,3     ; RETURN(1);
16
17 01103'024732 S115:   LDA     1       .SITEM   ; CHECKIDENT;
18 01104'000402         JMP             .+2      ; CHAIN:=ITEM; SKIP;
19 01105'024054 S101:   LDA     1       PROCESS  ; GETPROCESS;
20 01106'054733         STA     3       SLINK    ; CHAIN:=PROCESS;
21 01107'030713         LDA     2       S030     ; CURRENT:=CURR PARAM;
22 01110'151004         MOV     2,2     SZR      ; IF CURRENT<>0 THEN
23 01111'006010         SEARCHITEM               ; SEARCHITEM(CHAIN,CURRENT,ITEM);
24 01112'151004         MOV     2,2     SZR      ;
25 01113'010726         ISZ             SLINK    ;
26 01114'002725         JMP@            SLINK    ;
27
28 01115'030710 S17:    LDA     2       .S05     ; CLOSECURZONE;
29 01116'024711         LDA     1       .S07     ;
30 01117'146415         SUB#    2,1     SNR      ; IF CURZONE=INZONE THEN
31 01120'002220         .CLOSE                   ;  CLOSE(INZONE,TRUE);
32 01121'001400         JMP             0,3      ; RETURN;
33
34                S75:                           ; INT;
35 01122'004723         JSR             S11      ; SETZONE;
36 01123'004754         JSR             S113     ; CHECKEMPTY(
37 01124'000404         JMP             S755     ;  READ COMMANDS);
38 01125'004741         JSR             S16      ; SETIDENT;
39 01126'004755         JSR             S115     ; CHECKIDENT(REPEAT);
40 01127'000776         JMP             .-2      ;
41 01130'020060 S755:   LDA     0       FFIRST   ; BEGIN;
42 01131'030667         LDA     2       .S03     ;
43 01132'004430 S813:   JSR             S82      ; READ COMMANDS;
44 01133'020671         LDA     0       SENDC
45 01134'030666         LDA     2       S030     ;
46 01135'025001         LDA     1       +1,2     ;
47 01136'106405         SUB     0,1     SNR      ; IF NEXTCOM(1)=END THEN
48 01137'000405 .S801:  JMP             S801     ;  GOTO NEXTCOM1;
49 01140'020502         LDA     0       S04      ;
50 01141'000771         JMP             S813     ;
```

```
                    S80:                                ; NEXTCOMMAND:
01142'036657             LDA@     3        S03          ; FIRSTCOM:=FIRSTCOM(0);
01143'054656             STA      3        S03          ;
01144'034655  S801:      LDA      3        S03          ; NEXTCOM1:
01145'031403             LDA      2        +3,3         ; CUR:=FIRSTCOM(+3);
01146'050654             STA      2        S030         ;
01147'175004             MOV      3,3      SZR          ; IF FIRSTCOM<>0 THEN
01150'003401             JMP@              +1,3         ;  GOTO FIRSTCOM(+1);

01151'004744  S805:      JSR               S17          ; NEW COMMAND:
01152'020656             LDA      0        .S08         ; CLOSEINZONE;
01153'040654             STA      0        .S07         ; CURZONE:=OPZONE;
01154'024652             LDA      1        .S06         ;
01155'004700             JSR               S110         ; MOVENAME(PRIMARYIN,INZONE);
01156'020060             LDA      0        FFIRST       ;
01157'030641             LDA      2        .S03         ;
01160'004402             JSR               S82          ; READ COMMAND(LASTCORE,FIRSTCOMM
01161'000763             JMP               S801         ; GOTO NEXTCOM1;
```

```
| 0024  STM80
01                ; PROCEDURE READ COMMAND(PARAMTOP,NEXTCOMMAND);
02                ;         CALL:   RETURN:
03                ; AC0     PARAMTOP        PARAMTOP
04                ; AC1
05                ; AC2     NEXTCOMMAND
06                ; AC3     LINK            DESTROYED
07
08 01162'054657 S82:      STA     3       SLINK    ; READ COMMAND:
09 01163'040457           STA     0       S04      ; PARAMTOP:=PRAM(1);
10 01164'050636           STA     2       S030     ;
11 01165'030642           LDA     2       .S07     ; ZONE:=CURZONE;
12 01166'024651 S821:     LDA     1       .SWORK   ; REP:
13 01167'006223           INNAME                   ; INNAME(ZONE,WORK);
14 01170'020652           LDA     0       SWORK    ;
15 01171'101005           MOV     0,0     SNR      ; IF WORK(0)=0 THEN
16 01172'000774           JMP             S821     ;  GOTO REP;
17 01173'121000           MOV     1,0              ; SAVE(LASTCHAR);
18 01174'024637           LDA     1       .SCOMM   ;
19 01175'030642           LDA     2       .SWORK   ;
20 01176'006010           SEARCHITEM               ; SEARCHITEM(WORK,COMMANDS,ITEM);
21 01177'151005           MOV     2,2     SNR      ; IF ITEM=0 THEN
22 01200'000465           JMP             S900     ;  GOTO SYNTAX;
23 01201'031003           LDA     2       SIZE,2   ;
24 01202'034440           LDA     3       S04      ;
25 01203'051775           STA     2       -3,3     ; PARAMTOP(-3):=ITEM;
26 01204'126400           SUB     1,1              ;
27 01205'045774           STA     1       -4,3     ; PARAMTOP(-4):=0;
28 01206'045777           STA     1       -1,3     ; PARAMTOP(-1):=0;
29 01207'105000           MOV     0,1              ;
30 01210'004422           JSR             S8254    ; COUNTPARAM(4);
31 01211'052611           STA     2       @S030    ; NEXTCOMM(0):=PARAMTOP;
32 01212'050610           STA     2       S030     ; NEXTCOMM:=PARAMTOP;
33 01213'004417 S8220:    JSR             S8254    ; COUNTPARAM(4);
34 01214'020113 S822:     LDA     0       .32      ; WHILE
35 01215'122404           SUB     1,0     SZR      ;       LASTCHAR=SPACE DO
36 01216'002623           JMP@            SLINK    ;  BEGIN
37 01217'030610           LDA     2       .S07     ;
38 01220'024422           LDA     1       S04      ;  INNAME(CURZONE,PARAMTOP);
39 01221'006223           INNAME                   ;
40 01222'022420           LDA@    0       S04      ;
41 01223'101005           MOV     0,0     SNR      ;  IF PARAMTOP(0)=0 THEN
42 01224'000770           JMP             S822     ;   GOTO COUNTPAR;
43 01225'030415           LDA     2       S04      ;
44 01226'051007           STA     2       7,2      ; CHAIN.PREV:=CUR;
45 01227'102400           SUB     0,0              ;
46 01230'041003           STA     0       3,2      ; CHAIN.CUR:=0;
47 01231'000762           JMP             S8220    ;  END; RETURN;
48
49 01232'020145 S8254:    LDA     0       .M4      ;
50 01233'030407           LDA     2       S04      ; COUNTPARAM:
51 01234'113000           ADD     0,2              ;
52 01235'050405           STA     2       S04      ; PARAMTOP:=PARAMTOP-COUNT;
53 01236'020057           LDA     0       SNSTART  ;
54 01237'112513           SUBL#   0,2     SNC      ; IF PARAMTOP>=NSTART THEN
55 01240'001400           JMP             0,3      ;  RETURN;
56 01241'000413           JMP             S903     ; GOTO SIZE;
57
58 01242'000000 S04:      0       ; PARAMTOP
```

```
| 0025  STM80
01                  S909:                           ; BREAK ADDRESS:
02 01243'004427           JSR         S91           ;   TEXT:=
03                                                        .TXT    .BRK.
   01244'041122
   01245'045400
04
05                  S907:                           ; ILLEGAL:
06 01246'004426           JSR         S93           ;   TEXT:=
07                       .TXT                             .ILL.,  OR
   01247'044514
   01250'046000
08                  S904:                           ; CHECKSUM:
09 01251'004423           JSR         S93           ;   TEXT:=
10                       .TXT                             .SUM.,  OR
   01252'051525
   01253'046400
11                  S903:                           ; SIZE:
12 01254'004416           JSR         S91           ;   TEXT:=
13                       .TXT                             .SIZE.,  OR
   01255'051511
   01256'055105
   01257'000000
14                  S902:                           ; UNKNOWN:
15 01260'004435           JSR         S940          ;   TEXT:=
16                       .TXT                             .UNKNOWN.,  OR
   01261'052516
   01262'045516
   01263'047527
   01264'047000
17                  S900:                           ; SYNTAX:
18 01265'004425           JSR         S92           ;   TEXT:=
19                       .TXT                             .SYNTAX.;
   01266'051531
   01267'047124
   01270'040530
   01271'000000
20
21 01272'024417 S91:      LDA     1       SC805     ; RETURN:=NEW COMMAND;
22 01273'000402           JMP             .+2       ;
23 01274'024414 S93:      LDA     1       SC80      ; RETURN:=NEXT COMMAND;
24 01275'111000           MOV     0,2               ;
25 01276'161120           MOVZL   3,0               ;
26 01277'004425           JSR             S94       ; OUTCAUSE(TEXT);
27 01300'006211           OUTSPACE                  ;
28 01301'006216           OUTOCTAL                  ; OUTOCTAL(OUTZONE,VALUE);
29 01302'004446 S9035:    JSR             S97       ; ENDALARM: ENDMESSAGE;
30 01303'002402           JMP@            S9039     ; RETURN;
31
32 01304'000000 S9037:    0
33 01305'000000 S9039:    0
34 01306'001042'SCWORK:   SWORK
35 01307'001022'SCCUR:    S030
36 01310'001142'SC80:     S80
37 01311'001151'SC805:    S805
```

```
I 0026  STM80
01 01312'024777 S92:    LDA     1       SC805   ; RETURN:=NEW COMMAND;
02 01313'030773         LDA     2       SCWORK  ;
03 01314'000403         JMP             .+3     ;
04 01315'024773 S940:   LDA     1       SC80    ; RETURN:=NEXT COMMAND;
05 01316'032771         LDA@    2       SCCUR   ;
06 01317'161120         MOVZL   3,0             ;
07 01320'004404         JSR     S94             ; OUTCAUSE(TEXT);
08 01321'101120         MOVZL   0,0             ;
09 01322'004423         JSR             S96     ; CONTMESS(CUR PARAM);
10 01323'000757         JMP             S9035   ; GOTO END ALARM;
11
12 01324'044761 S94:    STA     1       S9039   ; OUTCAUSE:
13 01325'050757         STA     2       S9037   ;
14 01326'054404         STA     3       S949    ;
15 01327'004404         JSR             S95     ; FIRSTMESSAGE(TEXT);
16 01330'020754         LDA     0       S9037   ;
17 01331'002401         JMP@            S949    ;
18 01332'000000 S949:   0
```

```
| 0027  STM80
01 01333'054423 S95:    STA     3       S958    ; FIRSTMESS:
02 01334'040423         STA     0       S959    ;
03 01335'020121         LDA     0       .3      ;
04 01336'030422         LDA     2       SOPZONE ;
05 01337'006221         OPEN                    ; OPEN(OPZONE,3);
06 01340'020417         LDA     0       S959    ;
07 01341'024126         LDA     1       .10     ; CHAR:=10;
08 01342'006212 S951:   OUTCHAR                 ; OUTCHAR(OPZONE,CHAR);
09 01343'006215         OUTTEXT                 ; OUTTEXT(OPZONE,TEXT);
10 01344'002412         JMP@            S958    ;
11
12 01345'054411 S96:    STA     3       S958    ; CONTMESS:
13 01346'024113         LDA     1       .32     ;
14 01347'000773         JMP     S951            ;
15
16 01350'024130 S97:    LDA     1       .13     ; ENDMESS:
17 01351'054405         STA     3       S958    ;
18 01352'006214         OUTEND                  ;
19 01353'020120         LDA     0       .1      ;
20 01354'006221         OPEN                    ;
21 01355'002401         JMP@            S958    ;
22 01356'000000 S958:   0
23 01357'000000 S959:   0
24 01360'000251'SOPZONE:        S08
25
26
27              .TITLE AUTO
28
29              INZONE: S05                     ; INZONE ADDR
30              SGIV:   S059                    ; INZONE GIVEUP
31
32              S961:   LDA     2       INZONE  ;
33                      JSR             +0,3    ; ADDR:= NEW GIVEUP
34
35                      LDA     0       SGIV    ; INZONE.GIVEUP:= NORMAL GIVEUP;
36                      STA     0       ZGIVEUP,2;
37                      JMP@            .+1     ;
38                                      S805    ;  GOTO NEWCOMMAND;
39              .END
```

```
I 0028  STM80
01                  S0:                             ; PROCESS DESCRIPTOR:
02  01361'000000            0                       ;   NEXT
03  01362'000000            0                       ;   PREV
04  01363'000000            0                       ;   CHAIN
05  01364'000025            S2-S0                   ;   SIZE
06                  .TXT .S<0><0><0><0>.            ;   NAME
    01365'051400
    01366'000000
    01367'000000
07  01370'001370'           S0+EVENT                ;   FIRST EVENT
08  01371'001370'           S0+EVENT                ;   LAST EVENT
09  01372'001406'           S3                      ;   BUFFERS
10  01373'000000'           S4                      ;   PROGRAM
11  01374'000000            0                       ;   STATE
12  01375'000000            0                       ;   TIMER COUNT
13  01376'177777            -1                      ;   PRIORITY
14  01377'001243'           S909                    ;   BREAK ADDRESS
15  01400'001361'           S0                      ;   AC0
16  01401'000000            0                       ;   AC1
17  01402'001361'           S0                      ;   AC2
18  01403'000000            0                       ;   AC3
19  01404'003040"           S1000*2                 ;   STARTADD = SHOW S
20  01405'000000            0                       ;   SAVE
21                  S2:                             ; TOP OF PROCESS DESCRIPTOR:
22
23                  S3:                             ; MESSAGE BUFFER:
24
25  01406'001406'           .+0                     ;   NEXT
26  01407'001406'           .-1                     ;   PREV
27  01410'000000            0                       ;   CHAIN
28  01411'000012            BSIZE                   ;   SIZE
29  01412'001361'           S0                      ;   SENDER
30  01413'000000            0                       ;   RECEIVER
31          000004 .BLK  4                          ; MESS: 4 WORDS
32
33          001420'SLAST=.
34          000057 .LOC SNSTART
35  00057 001420'SLAST                              ;
36                  .NREL
37
38
39
```

```
| 0029  STM80
01
02                S1000:                                ;
03
04         000001 .IFN AUTO                             ;
05  01420'102400          SUB     0,0                  ; SHOW S: ! INIT CODE REMOVED AF
06  01421'004712          JSR             S95          ;   FIRSTMESS(0);
07  01422'004726          JSR             S97          ;   ENDMESS;
08  01423'002401          JMP@            .+1          ;  GOTO START;
09
10                .ENDC                                ;
11                .IFE SINDEV-2*SINDEV+AUTO-1          ;
12                      S75                            ;    IF INPUTDEVICE= PTR OR CDR TH
13                .ENDC                                ;       START = INT;
14
15         000017 .IFN SINDEV-2*SINDEV*AUTO            ;
16  01424'000672'         S500                         ;     IF INPUTDEVICE<> PTR OR CDR
17                .ENDC                                ;        START = CLEAR;
18
19                .IFE AUTO                            ;     IF AUTO START (MT,FD) THEN
20
21                                                     ;     BEGIN
22                                                     ;
23                      JSR             S961         ;
24                      STA     3       ZGIVEUP,2    ;    INZONE.GIVEUP:= RESTORE GIVE
25                      LDA     2       SOPZONE      ;
26                      LDA     1       .10          ;      OPZONE.ZREM:= 10;
27                      STA     1       ZREM,2       ;
28                      JMP@            .+1          ;       GOTO NEW COMMAND
29                                      S805         ;     END
30                .ENDC
31
32                ;   ******   END OF SYSTEM DRIVER   ******
33
34         001361'.END S0
```

```
S1000 : LDA     1     SOPSE     ;
        JSR           IN110     ;  FINDDRIVER (OPDEVICES, IDNAME)
        HALT                    ;  TO: ERROR
        LDA     2     S1100     ;  MOVENAME(IDNAME, OPZONE);
        JSR@          S1170     ;
        LDA     1     SDES      ;  FINDDRIVER(OPDEVICES, IDNAME);
        JSR           IN110     ;  TO: ERROR
        HALT                    ;
        LDA     2     S1110     ;  MOVENAME(IDNAME, PRIMIN);
        JSR@          S1170     ;
        LDA     2     S1120     ;  MOVENAME(IDNAME, INZONE);
        JSR@          S1170     ;
        MOV     1,3             ;
        LDA     0     +4,3      ;
        STA     0     ZSHAREL,2 ;  INZONE.SHAREL:=ID.LENGTH
        LDA     1     SNSTART   ;
        ADD     0,1             ;
                                  START OF CORE:= ENDOFS +
        STA     1     SNSTART   ;          ID.LENGTH
        STA@    1     S1150     ;
        LDA     0     +5,3      ;  INZONE.ZKIND:= ID.KIND;
        STA     0     ZKIND,2   ;  !CONTINUATOR := INT!
        LDA     1     .1013     ;
        AND     1,0   SZR       ;  IF ZKIND <> POSITIONABLE THEN
        JMP           S1003     ;  BEGIN
S1001 : SOB     0,0             ;    FIRSTMESS(0)
        JSR           S95       ;      ENDMESS;
        JSR           897       ;    GOTO CONTINUATOR
        JMP@          .+1       ;  END;
S1002 :               S75
```

```
S1003:  LDA   0     S1140   ;
        MOV   0,0   SZR     ;   NOT(
                                IF OUTO INT)  THEN GOTO ~~SKIP~~
        JMP         S1004   ;
        LDA   0     S1130   ;   ELSE
        STA   0     ZGLVECP,2   ;   BEGIN
        LDA   1     .10     ;       INZONE.ZREM := 10;
        STA   1     ZREM,2  ;
                                    GOTO NEXTCOMMAND
        JMP @       .+1     ;
                    S805    ;   END;

S1004:  LDA   0     S1160   ;
        STA   0     S1002   ;
        JMP         S1001   ;
```

```
S1100 :     S08     ;
S1110 :     S06     ;
S1120 :     S05     ;
S1130 :     S961    ;
S1140 :     AUTO    ;
S1150 :     .SLAST  ;
S1160 :     S500    ;
S1170 :     S110    ;
```

```
;       PROCEDURE    FIND
;
;               CALL          RETURN
;
;  AC0          -             DEST
;  AC1          CHAIN         NAMEADDR DEST
;  AC2          -             NAME ADDR
;  AC3          LINK          DEST
;
;  +0:    NOT FOUN
;  +1:    FOUND

IN100:          Ø                        ;
IN110:  STA     3       IN100            ;  FIND:
        STA     1       IN190            ;
IN120:  LDA     2       IN190            ;  < SUB  1,1
                                         ;    LDA  2   .1
                                         ;    WAITINTERRUPT
        INC     2,2                      ;
        LDA     1       IN180            ;
        SENDMESSAGE                      ;
        MOVLA   2,2     SZC              ;
        JMP             IN130            ;
        WAITANSWER                       ;
        MOVR    0,0     SZC              ;
        JMP             IN120            ;       IN180:  .+1
        LDA     1       IN190            ;                0
        INC     1,1                      ;                0
        ISZ             IN100            ;                0
        JMP@            IN100            ;                0
                                         ;       IN190:  Ø

IN130:  LDA@    2       IN190            ;
        STA     2       IN190            ;
        MOV     2,2     SZR              ;
        JMP             IN120            ;
        JMP@            IN100            ;
```

```
OPSEL:   [illegible] .+1          ;
         .+4                      ;   NEXT
         .TXT .TTY<0><0>.         ;
         0                        ;
         .TXT .OCP<0><0>.         ;


DSEL:    [illegible] .+1          ;
         .+6                      ;
         .TXT .MT<0><0><0>.       ;
         80                       ;
         14                       ;
         .+6                      ;
         .TXT .KIT<0><0>.         ;
         128                      ;
         14                       ;
         .+6                      ;
         .TXT .MTØ<0><0>.         ;
         80
         14                       ;
         .+6                      ;
         .TXT .PTR<0><0>          ;
         80
         1                        ;
         .+6                      ;
         .TXT .CDR<0><0>.         ;
         80                       ;
         2                        ;
         0                        ;
         .TXT .FDØ<0><0>.         ;
         128                      ;
         14                       ;
```