

EMS1 38/05-72

1. INTRODUCTION
2. SYNTAX
3. CONSTRUCTION OF THE EMS1 LANGUAGE
4. DESCRIPTION OF DEVICE TERMS
5. DESCRIPTION OF ENVELOPES AND GLISSANDI
6. COMMANDS
7. PSEUDOOPERATIONS
8. MACRO
9. OPERATING PROCEDURES

- APP. 1 PERMANENT SYMBOLS
- 2 STUDIO CONNECTIONS
 - 3 THE SORT RECORD
 - 4 ERROR PRINTOUTS AND MESSAGES
 - 5 ENVELOPE TABLES
 - 6 EXAMPLES OF EMS1 TEXT.
- .EJECT

1. INTRODUCTION

THE EMS1 LANGUAGE IS A SYMBOLIC LANGUAGE WHICH CAN BE USED TO CONTROL THE EMS STUDIO FROM ITS COMPUTER (PDP 15/40). THE EMS1 STATEMENTS SPECIFIES THE SETTING OF THE STUDIO PARAMETERS SUCH AS FREQUENCIES, AMPLITUDES, CONNECTIONS BETWEEN VARIOUS DEVICES IN THE STUDIO AND SO ON. THE INPUT TEXT MAY BE PRESENT ON ANY INPUT MEDIA EXCEPT MAGNETIC TAPE. THUS A USER MAY PUNCH HIS TEXT ON AN OFF-LINE TELETYPE AND THE RESULTING PAPERTAPE CAN BE READ IN BY THE PAPERTAPE READER ON THE COMPUT THIS TEXT IS TRANSLATED INTO THE INTERNAL DIGITAL FORM USED BY THE STUDIO (COMPILATION), I.E. THE EMS1 PROGRAMS CHECK THE INPUT TEXT FOR ERRORS AND PRODUCES OUTPUT IN THE FORM OF STUDIORECORDS ON MAGNETIC TAPE AND OUTPUT TEXT WITH THE LINES NUMBERED. IT IS POSSIBLE TO RUN THE EMS1 PROGRAMS INTERACTIVELY I.E. IT IS POSSIBLE TO PRODUCE A RECORD ON THE MAGNETIC TAPE, LISTEN TO THE RESULT AND IF NO GOOD MAKE CORRECTIONS AND ADDITIONS TO THE INPUT TEXT, LISTEN TO THE RESULT ONCE MORE AND IF GOOD SAVE THE RECORD LISTENED TO AND CONTINUE TO MAKE NEW ONES. THE LENGTH OF A PIECE IS UNLIMITED, THE USER JUST ADDS NEW RECORDS TO THE PREVIOUS ONES. THE LANGUAGE CONTAINS CONDITIONAL STATEMENTS WHICH MAKE IT POSSIBLE TO CONDITIONALLY CONTROL THE COMPILATION OF THE INPUT TEXT. THE "MACRO" OPTION MAKES IT POSSIBLE FOR THE USER TO REDUCE THE AMOUNT OF INPUT TEXT.

EMS1 STRUCTURE

THE SETUP OF THE STUDIO PARAMETERS THAT CAN BE DONE MANUALLY IN THE STUDIO CAN BE DONE IN SYMBOLIC FORM IN THE EMS1 LANGUAGE. THE USER CAN SPECIFY TO PRODUCING SOURCES (FREQUENCY GENERATORS, NOISE GENERATORS), THE SOUND^{Studio} MODIFYING DEVICES (FILTERS, RINGMODULATORS AND SO ON) AND THE CONNECTIONS BETWEEN THESE AND THE OUTPUT CHANNELS. TO THIS THE USER TO ALL AMPLIFIERS IN THE STUDIO CAN ADD ENVELOPE CURVES AND TO ALL FREQUENCY GENERATORS GLISSANDI CURVES. WHEN THE USER WRITES HIS INPUT TEXT, THE INFORMATION FROM THIS IS TRANSLATED TO SO CALLED "SORTRECORDS" (SEE. APP. 3) ON A DISK UNIT CALLED THE 'TEMP' DISK (TEMPORARY DISK UNIT). IT'S POSSIBLE TO LISTEN TO THE SOUNDS GENERATED FROM THESE SORTRECORDS WITH A 'PLAY' COMMAND. THE SOUNDS ON THE 'TEMP' DISK IS CALLED A SOUND OBJECT. WHEN THE USER IS SATISFIED WITH A SOUND OBJECT HE CAN TRANSFER IT TO ANOTHER DISK UNIT CALLED THE 'MIX' DISK WHICH CONTAINS THE SORTRECORDS FROM ONE OR MORE SOUND OBJECTS. THE INFORMATION ON THE MIX DISK IS CALLED A BLOCK. THIS MAY EVENTUALLY BE RECORDED ON MAGNETIC TAPE (MT) TOGETHER WITH A MT-'LABEL', WHICH MAKES IT POSSIBLE TO FIND THE BLOCK LATER ON. WITHIN A BLOCK A LOCAL TIME IS SPECIFIED, WICH STARTS WITH ZERO AT THE BEGINNING OF THE BLOCK. A SOUND OBJECT CAN BE TRANSFERRED TO THE MIX DISK WITH A LOCAL TIME SPECIFIED I.E. THE SOUND OBJECT IS TO START AT THIS TIME RELATIVE THE START OF THE MIX BLOCK START. A SOUND OBJECT CAN BE TRANSFERRED MORE THAN ONCE TO THE MIX DISK WITH DIFFERENT LOCAL TIMES EACH TIME. THE EMS1 TEXT MUST NOT BE WRITTEN IN TIME SEQUENCE. ^{the sound}

1.1 ORGANISATION OF A RUN

THE USE OF THE EMS1 SYSTEM CAN BE ARRANGED DIFFERENTLY DEPENDING ON

WHETHER IT IS INTERACTIVE OR NON-INTERACTIVE, AND WHETHER IT IS AN INITIAL OR CORRECTIVE RUN.

AN INITIAL RUN IS PERFORMED THE FIRST TIME THE INPUT TEXT IS PRESENTED TO THE COMPUTER. THE PROGRAM WILL PRODUCE BOTH A MAGNETIC TAPE WITH EMS CODE, USED TO PLAY THE PIECE ON THE STUDIO, AND AN OUTPUT TEXT, WHICH IS THE RECORDED INPUT TEXT WITH LINE NUMBERS INSERTED.

DURING A LATER CORRECTIVE RUN THE OUTPUT TEXT FROM THE PREVIOUS RUN MAY BE USED AS INPUT AND CORRECTIONS ADDED IN AN INTERACTIVE MODE. ALTERNATIVELY, THE EDIT UTILITY PROGRAM CAN BE USED TO INSERT ALTERATIONS BEFORE THE TEXT IS RECOMPILED. THE LINE NUMBERS ARE USED AS REFERENCE POINTS FOR FIND COMMANDS IN THE EDIT PROGRAM.

DURING AN INTERACTIVE RUN, TEXT IS TYPED IN AT THE DISPLAY TERMINAL (TV) OR POSSIBLY A TELETYPE, AND MESSAGES ABOUT ANY FORMAL ERRORS ARE RETURNED TO THE TERMINAL, WHICH IS THEN PLACED IN CORRECTIVE MODE. THE USER THEN IS EXPECTED TO CORRECT THE ERROR IMMEDIATELY. IT IS ALSO POSSIBLE TO HAVE THE LAST SETUP, OR ANY SEQUENCE OF SETUPS FROM THE PIECE, TO BE PLAYED BACK.

IN NON-INTERACTIVE MODE, AN ENTIRE SEQUENCE OF TEXT HAS BEEN PREPARED BEFOREHAND, EITHER BY PUNCHING A PAPER TAPE OFF LINE OR BY USE OF THE PIP AND EDIT UTILITY PROGRAMS. THE TEXT IS THEN COMPILED AT THE COMPUTER FROM BEGINNING TO END, AND ANY ERROR MESSAGES WILL BE LISTED ON THE LINE PRINTER OR A TELETYPE. THIS HAS THE ADVANTAGE THAT THE EMS STUDIO NEED NOT BE AVAILABLE AND THAT THE COMPUTER TIME USED IS CONSIDERABLY LESS. THE ERRORS CAN THEN BE CORRECTED BY USE OF THE EDIT PROGRAM OR DURING AN INTERACTIVE, CORRECTIVE RUN.

A NORMAL WAY TO WORK INTERACTIVELY COULD BE TO SPECIFY A SOUND OBJECT, LISTEN TO THE RESULT, IF NOT SATISFACTORY CHANGE SOME OF THE PARAMETERS, LISTEN TO AND SO ON UNTIL THE SOUND OBJECT IS SATISFACTORY. THE OBJECT CAN NOW BE ADDED TO THE PREVIOUS OBJECTS ON THE MIX DISK WITH A 'MIX' COMMAND. IT IS POSSIBLE TO LISTEN TO THE MIX BLOCK WITH A 'PLAY(MIX)' COMMAND. IT IS POSSIBLE TO OVERWRITE SORTRECORDS ON THE MIX DISK WITH SORTRECORDS FROM THE 'TEMP' DISK. THE RULE IS: IF TWO SORTRECORDS ARE GIVEN TO THE SAME DEVICE AT THE SAME LOCAL TIME, THE SORTRECORD INPUTTED LAST WILL BE KEPT AND THE OTHER(S) TAKEN AWAY. WHEN A MIX DISK BLOCK IS SATISFACTORY IT CAN BE RECORDED ON THE MT WITH AN 'END' COMMAND. NOTE THAT THE COMMANDS PLAY, PLAY(MIX) ALSO GENERATES CODE ON THE MT BUT CODE TO BE PERMANENTLY RECORDED IS RECORDED WITH THE 'END' COMMAND AND THUS ALWAYS FROM THE MIX DISK.
.EJECT

2. EHS1 SYNTAX

CONTENTS:

- 2.1 NOTATION
- 2.2 SYMBOLS
- 2.3 CHARACTER SET
- 2.4 CARRIAGE RETURN
- 2.5 COMMENTS
- 2.6 CONTEXT DEPENDENCE
- 2.7 SYNTAX

2.1 NOTATION

| METASYMBOL | STANDS FOR |
|------------|---|
| -> | IS DEFINED AS |
| / | OR (ALTERNATIVE SEPARATOR) |
| [N1=N2 | N1, N1+1, OR N2 EDITIONS OF THE CONTENTS BETWEEN THE BRACKETS |
| [N1=A | INFINITE N2 |
| [N | N1=N2=N |
| [] | N1=0, N2=1 |
| N1-N2 | INTEGER N, FOR WHICH $N1 \leq N \leq N2$ |
| TEXT | CHARACTER STRING NOT CONTAINING) OR (|

THE REST OF THE SINGLE CHARACTERS BETWEEN SPACES DENOTE THEMSELVES, AND THE REST OF THE CHARACTER STRINGS DENOTE COMPOSITE SYNTACTICAL OBJECTS AND CAN ALWAYS BE FOUND TO THE LEFT OF A ->.

2.2 SYMBOLS

THE SYMBOLS (LABELS, MACROS) ACCORDING TO 7.3, 7.12 AND 7.62 BELOW MAY NOT BE ONE OF THE FOLLOWING PERMANENT SYMBOLS:

A ALL AN AMP APP AT B BEGIN CALL CD CHA CLEAR DELETE END ENV
ERASE ESTEP EXIT FF FG FG3 FG6 FG9 FG12 FG15 FG18 FG21 FG24 FS GLIS
GSTEP IFDEF IFNEG IFPOS IFUND IFZER IN KEEP LIST LOOK LT MESS MEX MIX
NG NOLIST OLD PINK PLAY REPL REV RM SAVE SKIP STOTIM T TOP TRAPP TRY
WHITE WRITE Z

2.3 CHARACTER SET

DEFINE THE FOLLOWING BASIC CHARACTER SET:

BASIC-CHAR => LETTER / DIGIT / + / - / * / SLASH / = / . / , / ; /
: / (/) / < / > / # / & / " / \$ / % / ! / ? / ^ / @ /
CARRIAGE-RETURN

THEN THE FOLLOWING CHARACTER SET IS AVAILABLE FOR THE EMS1 TEXT:

EMS1-CHAR => BASIC-CHAR / ' !

THE ' FUNCTIONS AS A COMMENT SWITCH AND THE FOLLOWING CHARACTER SET
MAY BE USED IN COMMENTS (SEE 5 BELOW):

COMMENT-CHAR => BASIC-CHAR / SPACE / TAB

OTHER CHARACTERS NOT USED FOR SPECIAL PURPOSES (LIKE 'CONTROL C')
ARE IGNORED.

2.4 CARRIAGE RETURN AND RIGHT PARANTHESIS

CARRIAGE RETURN MAY BE USED

A/ AFTER) OR > OR # OR < OR ;

B/ INSTEAD OF ;

IN THE EMS1 TEXT AND ANYWHERE IN THE COMMENTS.

RIGHT PARANTHESIS) MAY ANYWHERE BE FOLLOWED BY SEMICOLON ; .
THE SAME RULES APPLY FOR THE MACRO END-DELIMITER " .

2.5 COMMENTS

COMMENTS CAN BE INSERTED ANYWHERE IN THE EMS1-TEXT AND HAVE THE FORMAT

COMMENT => ' [COMMENT-CHAR] 10-^ '

(SEE 3 ABOVE).

2.6 CONTEXT DEPENDENCE.

THE FOLLOWING CONTEXT DEPENDENCIES EXIST IN THE EMS1 LANGUAGE
AND ARE NOT FULLY MAPPED INTO THE CONTEXT-FREE SYNTAX BELOW:

1. A LABEL USED IN A PLAY COMMAND (7.16) MUST BE EARLIER DEFINED
IN A BLOCK (7.3)
2. A MACRO NAME (7.12) MUST BE EARLIER DEFINED IN A MACRO DEFINITION
(7.6)
3. A FILE (7.17) MUST BE SAVED BEFORE IT IS CALLED
WITH 'CALL' OR 'TOP' (7.7).
4. A VARIABLE SYMBOL MUST BE DEFINED IN AN ASSIGNMENT BEFORE IT IS USED IN
A TERM, COMMAND OR RIGHT HAND PART OF AN ASSIGNMENT
5. THE VALUES OF SEC,MS,FREQ,LEVEL ARE RESTRICTED AS FOLLOWS:

Q#SEC #16677 SECONDS
Q#MS #99999 MILLISECONDS
Q#FREQA#15999 HERTZ
Q#LEVEL#120 DECIBELS

6. THE PERMITTED CONNECTIONS (AND DISCONNECTIONS) ARE RESTRICTED
ACCORDING TO THE CURRENT STATE OF THE STUDIO. THIS ALSO HOLDS
FOR SOME OF THE HIGHER UNIT NUMBERS.

.EJECT

2.7 S Y N T A X

1. EMS1-PROGRAM => BLOCK [[K E E P TYPE] BLOCK]0-^
2. BLOCK => H E G T N BLOCK-PAR [ELEMENT / MACRO-DEF /
TEMP-COMMAND]1-^ E N D [BLOCK-PAR / ?]
3. BLOCK-PAR => (LABEL)
4. LABEL => SYMBOL
5. ELEMENT => COMMAND / CHAIN / ASSIGNMENT / CONDITIONAL / MACRO ; /
L T (TIME) [; / > CHAIN]1
6. MACRO-DEF => MACRO = " [ELEMENT / TEMP-COMMAND / M E X ;]1-^ " ;
7. TEMP-COMMAND => C A L L FILE / E R A S E FILE / I N LINES /
L O O K LINES / P L A Y PTYPE
R E P L FILE / S A V E FILE / S K I P LINES /
T O P FILE / T R A P P ; / T R Y [(TIME) / ;]1
8. COMMAND => A P P ; / C L E A R TYPE / D E L E T E LIST / K E E P TYPE /
H E S S (TEXT) / M I X [(TIME) / ;]1 /
[H O] L I S T ; / S T D T I M (M S) / W R I T E LIST
9. CHAIN => LINK [[> / *] LINK]0-^
10. ASSIGNMENT => SYMBOL = AREX ;
11. CONDITIONAL => [COND-HEAD COND-TEXT]1-^ <
12. MACRO => SYMBOL
13. TIME => [SEC] [, MS]
14. PTYPE => TYPE / (BLOCK-POINTER [, BLOCK-POINTER])
15. TYPE => (A L L) / (M I X) / ;
16. BLOCK-POINTER => LABEL
17. FILE => (SYMBOL [, EXT])
18. LIST => (SYMBOL [, SYMBOL]0-^)
19. LINK => GROUP [ENV-CHAIN] / POINT
20. AREX => [ADOP] PRIMARY [ADOP PRIMARY]0-^
21. COND-HEAD => I F [D E F / N E G / P O S / U N D / Z E R]1 (SYMBOL)
22. COND-TEXT => [ELEMENT]1-^
23. LINES => (POSNUM) / ;
24. ENV-CHAIN => [[>] ENV-LINK]1-^
25. TERM => AM-TERM / AMP-TERM / AT-TERM / CD-TERM / CHA-TERM / FF-TERM /
FG-TERM / NG-TERM / REV-TERM / RM-TERM / STEP-TERM
26. PRIMARY => SYMBOL / POSNUM
27. ADOP => + / =
28. GROUP => TERM [& TERM]0-^
29. ENV-LINK => ENV-TERM / GLIS-TERM / T-TERM / Z-TERM / STEP-TERM
30. EXT => [ALFANUM]3
31. POSNUM => [DIGIT]1-5 [. [DIGIT]0-5] / . [DIGIT]1-5
32. ENV-TERM => E N V (LEVEL , LEVEL , MS [. [FORM] [, MS]])
33. GLIS-TERM => G L I S (LEVEL , LEVEL , MS [, [FORM] [, MS]])
34. AM-TERM => A M ([M2] [, ENTRY [, LEVEL]])
35. AMP-TERM => A M P ([M2] [, LEVEL])
36. AT-TERM => A T ([M4] [, LEVEL])
37. CD-TERM => C D ([M5] [, [M4] [, LEVEL]])
38. CHA-TERM => C H A ([M4] [, LEVEL])
39. FF-TERM => F F ([M2] [, [FCNR] [, LEVEL]])
40. FG-TERM => F G ([FGNR] [, FREQ [, LEVEL [, WAVE]]])
41. NG-TERM => N G ([1] [, LEVEL [, COLOUR [, COLOUR]]])
42. REV-TERM => R E V ([M4] [, REVTIM [, LEVEL]])
43. RM-TERM => R M ([M3] [, ENTRY [, LEVEL]])
44. REVTIM => [SYMBOL / 1-15]
45. POINT => F S / F G POINTNR
46. LEVEL => [PRIMARY / O L D [ADOP PRIMARY]]
47. SEC => SYMBOL / 0-16677

- 48. MS => [SYMBOL / E-99999]
- 49. FREQ => [FREQSYM / O L D [ADOP FREQSYM]]
- 50. FORM => SYMBOL / -9-9
- 51. M2 => SYMBOL / 1 / 2
- 52. M3 => M2 / 3
- 53. M4 => M3 / 4
- 54. M5 => M4 / 5
- 55. FCNR => [SYMBOL / 1-28] [> [SYMBOL / 1-28]]
- 56. FCNR => [SYMBOL / 1-24] [> [SYMBOL / 1-24]]
- 57. ENTRY => [A / B]
- 58. FREQSYM => 0-15999 / SYMBOL
- 59. COLOUR => [W H I T E / P I N K]
- 60. WAVE => SYMBOL / 0-7
- 61. POINTNR => 3 / 6 / 9 / 1 2 / 1 5 / 1 8 / 2 1 / 2 4
- 62. SYMBOL => LETTER / [ALFANUM] 0-5
- 63. ALFANUM => LETTER / DIGIT
- 64. LETTER => A / B / C / D / E / F / G / H / I / J / K / L / M /
N / O / P / Q / R / S / T / U / V / W / X / Y / Z
- 65. DIGIT => 0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9
- 66. T-TERM => T (MS)
- 67. Z-TERM => Z
- 68. STEP-TERM => [E / G] 1 S T E P (MS)

NOTE TO 7.9:

A CHAIN ENDS WITH A ; IF THE LAST CHARACTER NOT IS A).
 .EJECT

3. EXPLANATION OF THE EMS1 LANGUAGE

3.1 DEFINITIONS

VALUES CAN BE OF TWO TYPES: INTEGER OR FRACTIONAL. AN INTEGER IS A NUMBER WITHOUT DECIMALS, A FRACTIONAL A NUMBER WITH AN INTEGER PART, A DECIMAL POINT AND A DECIMAL FRACTION. AN INTEGER MAY CONTAIN ONE TO FIVE DIGITS, A REAL MAY CONTAIN ONE TO FIVE DIGITS BOTH BEFORE THE DECIMAL POINT AND AFTER IT. EX. INTEGERS: 1 440 16000 FRACTIONALS: 80.25 40.5 FRACTIONALS ARE USED TO SPECIFY QUARTER OF DECIBELS. THEREFORE THE FRACTIONAL PART IS ALWAYS ROUNDED TO THE NEAREST QUARTER INTEGER.

EX. 1.20235 WILL BE ROUNDED TO 1.25

1.1 WILL BE ROUNDED TO 1.0

A SYMBOL CONTAINS ONE TO SIX CHARACTERS FROM THE FOLLOWING SET:

THE LETTERS A THROUGH Z

THE DIGITS 0 THROUGH 9

THE FIRST CHARACTER OF A SYMBOL MUST BE A LETTER

EX. LEGAL SYMBOLS: DB SUMR1 VIA

ILLEGAL SYMBOLS: 100DB .AB *SYM

3.2 PERMANENT SYMBOLS

THE EMS1 LANGUAGE USES A SET OF RESERVED SYMBOLS WHICH HAVE A SPECIAL MEANING TO THE EMS1 SYSTEM. THESE SYMBOLS ARE DEFINED BY THE EMS1 PROGRAMS WHEN THE SYSTEM IS LOADED AND CAN NOT BE CHANGED OR USED FOR OTHER PURPOSES THAN IS DESCRIBED IN THIS MANUAL.

A COMPLETE LISTING OF THE PERMANENT SYMBOLS IS FOUND IN APP. 2

THE MEANING OF THE SYMBOLS ARE DISCUSSED IN CHAPTERS 4-8.

3.3 ASSIGNMENT OF VALUES TO SYMBOLS

IT IS POSSIBLE TO ASSIGN A NUMBER TO A SYMBOL IN THE FOLLOWING WAY:

EX. GIVE THE SYMBOL 'SYM' THE VALUE 20

WRITE: SYM=20

GIVE THE SYMBOL 'DB' THE VALUE 50.75

WRITE: DB=50.75

SYMBOLS WITH ASSIGNED VALUES ARE CALLED VARIABLES. VARIABLES MAY BE USED ANYWHERE IN THE TEXT INSTEAD OF A NUMBER.

EX. THE FOLLOWING TWO LINES WILL PRODUCE THE SAME RESULT:

FG(1,1000,100,1)

NR=1;FR=1000;FG(NR,FR,100,1)

A VARIABLE MAY BE CHANGED FROM ONE VALUE TO ANOTHER ANYWHERE IN THE TEXT. EXCEPTION: IF A VARIABLE IS ASSIGNED IN THE FOLLOWING WAY IT IS NOT POSSIBLE TO CHANGE.

EX. SYM==2

I.E. IF TWO EQUAL SIGNS ARE USED INSTEAD OF ONE A VARIABLE WILL BE MARKED PERMANENT, AND ANY FURTHER ATTEMPT TO MODIFY IT WILL BE NOTED AS AN ERROR.

EX. NR=10;FR==440;FG(NR,FR,100,2) NR=11;FG(NR,FR,100,2)

NR IS FIRST ASSIGNED TO '10' SO THE FIRST TIME 'NR' IS USED IT HAS THE VALUE '10' BUT THEN THE VALUE OF NR IS CHANGED TO '11' SO THAT THEREAFTER THE VALUE OF 'NR' IS '11' UNTIL IT IS CHANGED AGAIN. A STATEMENT LIKE: FR=880 WOULD HAVE NO EFFECT BECAUSE 'FR' CANNOT BE CHANGED IN THIS CASE.

3.4 ARITHMETIC EXPRESSIONS

IN ADDITION TO THE POSSIBILITY TO ASSIGN A VALUE ,WHICH IS A SINGLE NUMBER, TO A SYMBOL IT IS ALSO POSSIBLE TO ASSIGN A VALUE WHICH IS THE THE RESULT OF AN ARITHMETIC EXPRESSION. THE VALUES MAY BE NUMBERS AND SYMBOLS MIXED. THE ARITHMETIC OPERATORS ARE: + - * AND /.

EX. V1=10;V2=20; V3=30; VAL=10+V1+V2-V3
THIS GIVES 'VAL' THE VALUE '10' IN THIS CASE.

EX. NR=NR+1

THIS ADDS '1' TO THE VALUE OF NR. NOTE THAT THE VALUE OF 'NR' IS NOT CHANGED UNTIL THE WHOLE EXPRESSION IS EVALUATED.

AN ERROR CONDITION OCCURS IF A VARIABLE TO THE RIGHT OF THE EQUAL SIGN IS NOT DEFINED EARLIER IN THE TEXT.

THE VARIABLES TO THE RIGHT OF THE EQUAL SIGN REMAIN UNCHANGED DURING THE EVALUATION OF AN EXPRESSION.

A SYMBOL WILL GET THE SAME VALUETYPE AS THE FIRST VALUE TO THE RIGHT OF THE EQUAL SIGN.

EX. SYM=10.25+5

'SYM' WILL BE A FRACTIONAL VARIABLE

EX. SYM1=5+10.25

SYM1 WILL BE AN INTEGER VARIABLE.

EX SYM2=VAR +17-NR

THE VARIABLE 'SYM2' WILL BE OF THE SAME TYPE AS 'VAR'

THE MAXIMUM MAGNITUDE OF AN INTEGER IS 131071 AND OF A FRACTIONAL 32767.

3.5 ASSIGNMENT OF A TEXTSTRING TO A SYMBOL

A WAY TO REDUCE THE AMOUNT OF INPUT TEXT IS TO USE THE STRING ASSIGNMENT FEATURE. AN ASSIGNMENT IS DONE IN THE FOLLOWING WAY:

EX. SOUND="FG(1,FREQ,,2)>CHA(1,100)"

THE TEXT WITHIN THE TWO " IS ASSIGNED TO THE SYMBOL 'SOUND'

WHENEVER THE SYMBOL 'SOUND' IS FOUND IN THE INPUT TEXT

THE ASSIGNED TEXT IS COPIED AND COMPILED.

THE ASSIGNED TEXT MAY CONSIST OF MORE THAN ONE LINE.

A SYMBOL TO WHICH A TEXT STRING IS ASSIGNED IS CALLED A 'MACRO'

THE MACRO IS DISCUSSED IN DETAIL IN CHAPTER 8.

3.6 COMMENTS

COMMENTS MAY APPEAR ANYWHERE IN THE TEXT ENCLOSED IN QUOTES (').

THE ONLY THING THAT IS DONE ABOUT COMMENTS IS THAT THEY ARE TRANSFERRED TO THE OUTPUT TEXT BUT THE COMPILATION IS NOT AFFECTED.

EXCEPTION: A COMMENT STARTING IN POSITION 1 OF A LINE WILL BE TOTALLY IGNORED.

EX. 'THIS IS A COMMENT'

3.7 DELIMITERS.

DELIMITERS ARE SPECIAL CHARACTERS USED TO SEPARATE VARIABLES ,NUMBERS AND OTHER SYMBOLS IN THE TEXT. THEY ARE LISTED IN APP. 3

BLANKS AND TABS MAY BE INSERTED ANYWHERE TO INCREASE

READABILITY BUT THEY ARE TREATED LIKE COMMENTS.

THE FUNCTION OF THE OTHER DELIMITERS ARE DISCUSSED IN THE FOLLOWING CHAPTERS.

3.8 ELEMENTS AND TERMS

AN ELEMENT IS A NUMBER OR SYMBOL (NOT COMMENT OR BLANKS OR TABS) BETWEEN TWO DELIMITERS.

AN ELEMENT MAY BE AN EMPTY ELEMENT IF THERE IS NO NUMBER OR SYMBOL BETWEEN TWO DELIMITERS.

EX. FG(NR, 'ALREADY SET', 100, 5)

FIRST ELEMENT: FG, SECOND ELEMENT: NR, THIRD ELEMENT: EMPTY
A PERMANENT SYMBOL WITH ITS PARAMETERS IS CALLED A TERM.

EX. FG(1, 440, 50, 3) RM(1, A, 100) CHA(1, 100) FG(1)

A TERM MUST BE WRITTEN ON ONE LINE ONLY.

3.9 OUTPUT TEXT

THE OUTPUT TEXT CONSISTS OF THE INPUT TEXT, ERROR CORRECTIONS (SEE 3.5), AND MACRO EXPANSIONS (SEE CHAPTER 8) AND LINE NUMBERING.

THE LINE NUMBER FORMAT IS THE FOLLOWING:

| POS | CONTENTS |
|-----|-------------|
| 1 | ' |
| 2-6 | LINE NUMBER |
| 7 | ' |
| 8 | BLANK |

EX. INPUT TEXT: FG(NR, F440, DB, WAVEF) >
OUTPUT TEXT: '00005' FG(NR, F440, DB, WAVEF) >

3.10 ERROR CORRECTION

IF EMS1 IS RUN IN INTERACTIVE MODE ERRORS CAN BE CORRECTED IN THE FOLLOWING WAY:

EX. INPUT TEXT: N=18; FG(N, FR, 100, 1) > RM(1, A, 100)

WHERE THE SYMBOL 'FR' IS NOT DEFINED.

ERROR PRINTOUT: FG(N, FR)
04 001 SYMBOL NOT DEFINED

THE START OF THE FIRST LINE 'FG' DENOTES FROM WHERE THE CORRECTION SHOULD BE TYPED IN, AND THE END 'FR' WHERE THE ERROR WAS DISCOVERED. FINALLY ')' DENOTES THAT AFTER THE CORRECTION HAS BEEN ACCEPTED, TRANSLATION IS RESUMED AFTER THE RIGHT PARANTHESIS WITH '>RM(1, A, 100)'.

AFTER THE ERROR MESSAGE A ! IS PRINTED OUT TO TELL THE USER THAT HE IS IN CORRECTION MODE. USER MAY NOW INSERT CHANGES INSTEAD OF THE THE ERRONEOUS TERM.

THE USER WRITES: FR=880; FG(N, FR, 100, 1)

THE ANSWER FROM EMS1 IS ANOTHER ! .IT IS NOW POSSIBLE TO INSERT MORE CORRECTIONS OR ADDITIONS TO THE INPUT TEXT. TO RETURN FROM CORRECTION MODE MAKE A CARRIAGE RETURN ALONE. THE INPUT WILL NOW CONTINUE FROM THE CHARACTER AFTER THE ERRONEOUS TERM. IN THIS CASE: >RM(1, A, 100)

THE THE RESULTING OUTPUT TEXT FROM THIS CORRECTION:

'00001' N=18
'00002' FR=880; FG(N, FR, 100, 1)
'00003' >RM(1, A, 100)

.EJECT

4. DESCRIPTION OF LT AND DEVICE TERMS

LT LOCAL TIME

LT TERM IS OF FORM

LT(SEC,MS)

SEC TIME IN SECONDS FOR LOCAL TIME.

MS TIME IN MILLISECOND FOR LOCAL TIME

IF ONLY ONE PARAMETER IS GIVEN MILLISECONDS IS ASSUMED.

EXAMPLES

LT(1,1000) 'LOCAL TIME 1 SEC. 1000 MS= 2 SEC.'

LT(2,) 'LOCAL TIME 2 SEC.'

LT(500) 'LOCAL TIME 500 MS'

FG FREQUENCY GENERATORS

FG-TERM IS OF FORM

FG(NR,FREQ,INTENS,WAVEF) WHERE

NR FREQUENCY GENERATOR NUMBER (1-24)

FREQ FREQUENCY (1-15999)

INTENS AMPLITUDE IN DB (0-120)

WAVEF WAVEFORM (0-7)

MORE THAN ONE 'FG' MAY BE GIVEN VALUES IN THE SAME FG TERM. WRITE:

FG(NR1>NR2,FREQ,INTENS,WAVEF) WHERE

NR1,NR2 FREQUENCY GEN. NR (1-24)

NR1 .LE. NR2

ALL PARAMETERS EXCEPT SOUND GENERATOR NUMBER MAY BE OMITTED IF THE SOUND GENERATOR NR HAS BEEN USED EARLIER IN THE PIECE.

AN OMITTED PARAMETER WILL GET THE VALUE USED AT THE LATEST CALL FOR THAT GENERATOR

EXAMPLE

FG(1,440,80,2)

FG(2,550,80,3)

FG(1,,90,2)

HERE SOUND GENERATOR ONE WILL STILL HAVE FREQUENCY 440 HERTZ

NR, FREQ AND WAVEF MUST BE INTEGERS WHILE INTENS COULD BE INTEGER OR FRACTIONAL

EXAMPLE

FG(1,440,80,2)

FG(2,550,80,5,3)

THE NUMBER OF DECIMALS IN INTENS MAY NOT EXCEED 5.
THE FRACTIONAL PART WILL BE CONVERTED TO THE
NEAREST QUARTER DECIBEL VALUE

NOTE

ALL SOUND GENERATORS' INTENSITY WILL BE SET TO ZERO
EVERY TIME A 'NEXT' COMMAND IS ISSUED.
THIS COULD BE OVERRIDED BY USING THE 'DITTO' TERM

NG NOISE GENERATOR

NG-TERM IS OF FORM

NG(NR,INTENS,COL1,COL2) WHERE

NR NUMBER OF NOISE GENERATOR (ONLY 1 IS POSSIBLE AT PRESENT)
INTENS AMPLITUDE IN DB (0-120)
COL1,COL2 NOISE COLOUR (WHITE OR PINK)

BOTH PARAMETERS MAY BE OMITTED IF THE NG-TERM IS USED
EARLIER IN THE PIECE
IF THE COLOUR IS OMITTED THE NOISE COLOUR WILL REMAIN
THE SAME AS BEFORE AND IF THE INTENSITY IS OMITTED
ONLY THE NOISE COLOUR WILL BE AFFECTED
IF BOTH PARAMETERS ARE OMITTED THE TERM WILL CAUSE
NO OPERATION IF IT IS NOT PRECEDED OR SUCCEDED
BY '>' OR '#'

COL1,COL2 MUST TAKE VALUE WHITE OR PINK
INTENS MAY BE INTEGER OR REAL

EXAMPLE

NG(1,100,WHITE,PINK)
NG(1,,PINK)

THE NUMBER OF DECIMALS IN THE INTENSITY MAY NOT EXCEED 5.
THE FRACTIONAL PART WILL BE CONVERTED TO THE NEAREST
QUARTER DECIBEL VALUE

NOTE

IF THE NOISE GENERATOR SHOULD BE SET TO BOTH WHITE
AND PINK NOISE THIS MUST BE DONE WITH THE EX-TERM
THE USE OF NG-TERM WITH COLOUR WHITE WILL ALWAYS
CLEAR THE COLOUR PINK AND VICE VERSA

FF FILTER UNITS

FF-TERM IS OF FORM

FF(FNR,FCH,INTENS) WHERE
FNR FILTER NUMBER (1-2)
FCH FILTER CHANNEL (1-26)
INTENS AMPLITUDE IN DB (0-120)

MORE THAN ONE FILTER CHANNEL MAY BE GIVEN VALUES IN THE SAME 'FF' TERM. WRITE:

FF(FNR,FCH1>FCH2,INTENS) WHERE

FCH1,FCH2 FILTER CHANNEL NR (1-28)
FCH1 .LE. FCH2

ALL PARAMETERS MAY BE OMITTED IF THE FILTER TERM HAS BEEN USED EARLIER IN THE PIECE
IF THE FILTER NUMBER IS OMITTED FNR WILL TAKE THE SAME VALUE AS IN THE LAST FF-TERM
IF THE CHANNEL NUMBER IS OMITTED FCH WILL TAKE THE SAME VALUE AS IN THE LAST FF-TERM (REGARDLESS OF FILTER NUMBER)
IF THE INTENSITY IS OMITTED THE TERM WILL CAUSE NO OPERATION ON THE FILTER BUT COULD CAUSE A CONNECTION OR DISCONNECTION IF IT IS PRECEDED OR SUCCEEDED BY '>' OR '#'

FNR AND FCH MUST BE INTEGERS WHILE INTENS COULD BE INTEGER OR REAL

EXAMPLE

FF(1,13,80)
FF(1,14,80.5)

THE NUMBER OF DECIMALS IN INTENS MAY NOT EXCEED 5. THE FRACTIONAL PART WILL BE CONVERTED TO THE NEAREST QUARTER DECIBEL VALUE

REV

REVERBERATION UNITS

REV-TERM IS OF FORM

REV(RU,RTIM,INTENS) WHERE
RU REVERBERATION UNIT (1-2)
RTIM REVERBERATION TIME (1-15) APPROX. RTIM*0.35 SEC.
INTENS AMPLITUDE IN DB (0-120)

ALL PARAMETERS MAY BE OMITTED IF THE REVERBERATION TERM HAS BEEN USED EARLIER IN THE PIECE
IF THE REVERBERATION UNIT IS OMITTED RU WILL TAKE THE SAME VALUE AS IN THE LAST REV-TERM
IF THE REVERBERATION TIME IS OMITTED RTIM WILL TAKE THE SAME VALUE AS IN THE LAST REV-TERM (REGARDLESS OF REVERBERATION UNIT)
IF THE INTENSITY IS OMITTED THE TERM WILL CAUSE NO OPERATION ON THE REVERBERATION UNIT BUT COULD CAUSE A CONNECTION OR DISCONNECTION IF THE TERM IS PRECEDED OR SUCCEEDED BY '>' OR '#'

RU AND RTIM MUST BE INTEGERS WHILE INTENS COULD BE INTEGER OR REAL

EXAMPLE

REV(1,10,80)
REV(2,,70.5)

RM RING MODULATORS

RM-TERM IS OF FORM

RM(NR,ENTRY,INTENS) WHERE
NR RING MODULATOR UNIT NUMBER (1-3)
ENTRY ENTRY (A OR B)
INTENS AMPLITUDE IN DB (0-120)

ALL PARAMETERS MAY BE OMITTED IF THE RINGMODULATOR TERM
HAS BEEN USED EARLIER IN THE PIECE
IF THE RING MODULATOR NUMBER IS OMITTED NR WILL
TAKE THE SAME VALUE AS IN THE LAST RM-TERM
IF THE RING MODULATOR ENTRY IS OMITTED ENTRY WILL
TAKE THE SAME VALUE AS IN THE LAST RM-TERM
(REGARDLESS OF RINGMODULATOR UNIT)
IF THE INTENSITY IS OMITTED THE TERM WILL CAUSE NO
OPERATION ON THE RINGMODULATOR UNIT IN QUESTION
BUT COULD CAUSE A CONNECTION OR DISCONNECTION IF IT IS
PRECEDED OR SUCCEEDED BY '>' OR '#'

NR MUST BE INTEGER
ENTRY MUST TAKE VALUE 'A' OR 'B'
INTENS MAY BE INTEGER OR REAL

EXAMPLE

RM(1,A,80)
RM(2,B,75.750)

THE NUMBER OF DECIMALS IN THE INTENSITY MAY NOT EXCEED 5
THE FRACTIONAL PART WILL BE CONVERTED TO THE NEAREST
QUARTER OF DECIBEL

AM AMPLITUDE MODULATORS

AM-TERM IS OF FORM

AM(NR,ENTRY,INTENS) WHERE
NR AMPLITUDE MODULATOR NUMBER (1-2)
ENTRY ENTRY (A OR B)
INTENS AMPLITUDE IN DB (0-120)

ALL PARAMETERS MAY BE OMITTED IF THE AMPLITUDE MODULATOR TERM
HAS BEEN USED EARLIER IN THE PIECE
IF THE AMPLITUDE MODULATOR NUMBER IS OMITTED NR WILL
TAKE THE SAME VALUE AS IN THE LAST AM-TERM
IF THE AMPLITUDE MODULATOR ENTRY IS OMITTED ENTRY
WILL TAKE THE SAME VALUE AS IN THE LAST AM-TERM
(REGARDLESS OF AMPLITUDE MODULATOR NUMBER)
IF THE INTENSITY IS OMITTED THE TERM WILL CAUSE NO
OPERATION ON THE RINGMODULATOR IN QUESTION
BUT COULD CAUSE A CONNECTION OR DISCONNECTION IF THE

TERM IS PRECEDED OR SUCCEDED BY '>' OR '#'

NR MUST BE INTEGER

ENTRY MUST TAKE VALUE 'A' OR 'B'

INTENS MAY BE INTEGER OR REAL

EXAMPLE

AM(1,A,80)
AM(2,B,70.25)

THE NUMBER OF DECIMALS IN THE INTENSITY MAY NOT EXCEED 5
THE FRACTIONAL PART OF THE INTENSITY WILL BE CONVERTED
TO THE NEAREST QUARTER DECIBEL.

AMP AMPLIFIERS

AMP TERM IS OF FORM

AMP(NR,INTENS) WHERE
NR AMPLIFIER NUMBER (1 OR 2)
INTENS AMPLITUDE IN DB (0-120)

EXAMPLE

AMP(1,100)
AMP(2,80.5)

CD CHANNEL DISTRIBUTORS

CD TERM IS OF FORM

CD(NR,CHNR,INTENS) WHERE
NR 'CD' NUMBER (AT PRESENT ONLY 1 IS POSSIBLE)
CHNR CHANNEL NR (1-4)
INTENS AMPLITUDE IN DB (0-120)

EXAMPLE

CD(1,3,100)

OLD THE 'OLD' SYMBOL

OLD, OLD+VALUE, OLD-VALUE MAY BE USED INSTEAD OF NORMAL
FREQUENCY OR AMPLITUDE VALUES. THE VALUE OF 'OLD' IS SET
BY THE CODEGENERATING PROGRAM TO THE LAST VALUE OF THE REFERRED
PARAMETER.

EXAMPLE

LT(10)FG(1,OLD-10,OLD+10)
LT(5) FG(1,9000,80,5)

IN THIS CASE THE OLD VALUE WILL BE 9000 FOR THE
FREQUENCY AND 80 FOR THE AMPLITUDE AT LOCAL TIME 10 SEC.

CONNECT CONNECTIONS AND DISCONNECTIONS

TO MAKE A (DIS-)CONNECTION BETWEEN TO DEVICES IN THE STUDIO
WRITE THE DELIMITER > (#) BETWEEN THE CORRESPONDING DEVICE TERMS.

EXAMPLE

FG(1,200)>CHA(1,100)
FG3#CHA(2)

THERE ARE SOME SPECIAL SYMBOLS FOR CONNECTION POINTS
FG3,FG6 ETC (SEE APP. 1). THESE MAY BE USED INSTEAD OF DEVICE TERMS
TO MAKE CONNECTIONS

EXAMPLE

FG3>FG6>FG9>FG12
FG(20)>FS

DEVICE CHAINS

A DEVICE CHAIN CONTAINS DEVICE TERMS WITH THE DELIMITER & BETWEEN
EACH TERM.

EXAMPLES

FG(1)&FG(7>10)&FF(1,1>7,100)
CHA(1)&CHA(2,100)&CHA(4,80)

A DEVICE CHAIN MAY BE CONNECTED TO ANOTHER DEVICE CHAIN OR TERM.

EXAMPLE

FG(1,1000,100,1)&FG(4,800,80,2)&FG(10)>CHA(1,100)&CHA(2,100)
THIS IS EQUIVALENT TO WRITING:
FG(1,1000,100,1)>CHA(1,100)
FG(1)>CHA(2)
FG(4,800,80,2)>CHA(1)
FG(4)>CHA(2,100)

.EJECT

5. DESCRIPTION OF ENVELOPE AND GLISSANDI TERMS.

ENV ENVELOPE DESCRIPTION

ENV-TERM IS OF FORM
ENV(AMP1,AMP2,T,TYPE,STEP) WHERE

AMP1 START INTENSITY IN DB (0-120)
AMP2 END INTENSITY IN DB (0-120)
T TIME IN MILLISECONDS
TYPE ENVELOPETYPE (-9 - 9)
STEP STEP IN MILLISECONDS
IF 'STEP' IS OMITTED THE ENV STEP WILL BE THAT IN PREVIOUS
'ESTEP' TERM. (SEE BELOW)

THE INTENSITY OF THE ASSOCIATED AMPLIFIER WILL
BE VARIED FROM AMP1 DB TO AMP2 DB IN T MILLISECONDS.
THE WAY THE INTENSITY IS VARIED IS CONTROLLED BY TYPE
WHICH SPECIFIES ONE OUT OF 19 POSSIBLE WAYS.

TYPE=0 LINEAR VARIATION
-9 < TYPE < 0 CONCAVE VARIATION
0 < TYPE < 9 CONVEX VARIATION
THE EXACT SHAPE OF THE VARIATION IS FOUND IN APPENDIX 5.

EXAMPLES

ENV(50,100,500,5,10)
ENV(80,100,100,1)

T TIME OF DURATION

THE T TERM IS OF FORM

T(TIME) WHERE

TIME TIME IN MILLISECONDS

T(TIME) IS EQUIVALENT TO: ENV(OLD,OLD,TIME)>Z

Z Z TERM IS OF FORM

Z

Z IS EQUIVALENT TO: ENV(OLD,0,10,0,10)

ESTEP STEP OF VARIATION ESTEP-TERM IS OF FORM

ESTEP(T)

T TIME IN MILLISECONDS.
IF ESTEP NOT SPECIFIED BY USER,ESTEP WILL BE SET TO 1.
THE FUNCTION OF ESTEP IS BEST EXPLAINED BY EXAMPLES

EXAMPLES

ESTEP(2)
ESTEP(35)

ENVELOPE ON SOUNDGENERATOR 1 WITH START INTENSITY 90 DB,
END INTENSITY 100 DB, TIME 20 MILLISECONDOS AND
TYPE W (LINEAR ENVELOPE).

A) ESTEP=1 (IF NOT SPECIFIED WILL BE ESTEP=1)
THE INTENSITY OF SOUNDGENERATOR 1 WILL BE CHANGED
IN STEPS OF 1/2 DB, EACH STEP WILL REMAIN UNCHANGED
FOR 1 (ESTEP) MILLISECOND AND THERE WILL BE 20
STEPS GENERATED.

B) ESTEP=2
THE INTENSITY OF SOUNDGENERATOR 1 WILL BE CHANGED
IN STEPS OF 1 DB, EACH STEP WILL REMAIN UNCHANGED
FOR 2 (ESTEP) MILLISECONDOS AND THERE WILL BE 10
STEPS GENERATED.

THE VALUE OF ESTEP ALSO AFFECTS ALL OTHER TIMES IN THE
ENVELOPE SPECIFICATION IN THAT THE TIMES MUST ALL
BE EQUAL TO OR GREATER THAN ESTEP AND THAT THE NUMBER OF
INTENSITY CHANGES GENERATED BY AN ENV-TERM ,T/ESTEP,
IS CALCULATED IN INTEGER ARITHMETIC.
THE NUMBER OF INTENSITY CHANGES GENERATED BY ONE ENV-TERM
(T/ESTEP) MUST BE GREATER THAN 0.
THE ESTEP-TERM MAY BE WRITTEN AT ANY PLACE IN THE
ENVELOPE-SPECIFICATION ,IF OMITTED ESTEP IS SET TO 1.

ENVELOPE-CHAIN

AN ENVELOPE CHAIN CONSISTS OF ENV,T AND Z TERMS OPTIONALLY
SEPARATED BY THE DELIMITER >. THE CHAIN IS TERMINATED
BY A NONE ENVELOPE TERM OR A ;

EXAMPLES

ENV(50,100,100,1)>ENV(100,80,100)>T(1000)
ENV(100,80,10,1,1)>ENV(80,100,1000,2,10)>Z

AN ENVELOPE CHAIN CAN BE CONNECTED TO A SINGLE DEVICE
OR A GROUP OF DEVICES BY CONNECTING THE ENV-CHAIN TO
A DEVICE TERM RESP. A DEVICE CHAIN.

EXAMPLE

FG(1,440)>ENV(50,100,10000)>ENV(100,30,100)
FG(1>3)&FG(7)>ENV(50,100,1000,1)>Z

IT IS POSSIBLE TO CONNECT A TERM OR DEVICE CHAIN TO BOTH
AN ENV-CHAIN AND ANOTHER DEVICE CHAIN.

EXAMPLE

FG(1)&FG(4)>ENV(30,80,1000)>ENV(80,90,1000)>CHA(1,100)&CHA(2,100)

NOTE THAT THE DELIMITER > HAS THE MEANING: "CONNECT THE DEVICE

(OR DEVICE CHAIN)" IF IT IS WRITTEN BEFORE A DEVICE TERM.
THE > BEFORE THE ENV TERMS MAY BE OMITTED.

GLIS GLISSANDO DESCRIPTION

THE GLIS TERM IS OF FORM

GLIS(FREQ1,FREQ2,TIME,TYPE,STEP) WHERE

FREQ1 START FREQUENCY IN HZ (1-15999)
FREQ2 END FREQUENCY IN HZ (1-15999)
TIME TIME OF DURATION IN MS.
TYPE GLISSANDO TYPE (-9 TO +9)
STEP GLISSANDO STEP
(SEE DESCRIPTION OF STEP IN ENV TERM)

GSTEP STEP OF VARIATION IN GLIS TERMS

GSTEP TERM IS OF FORM

GSTEP(T) WHERE

T TIME OF STEP IN MS.
(SEE DESCRIPTION OF ESTEP)

GLISSANDO CHAIN

THE GLISSANDO CHAIN IS CONSTRUCTED IN THE SAME WAY AS THE
ENVELOPE CHAIN.

EXAMPLE

GLIS(80,800,1000,1,10)>GLIS(800,1000,100,2)>GLIS(1000,800,100)

ENV AND GLIS TERMS CAN BE MIXED IN THE SAME CHAIN SO THAT A
ENV AND GLIS CHAIN CAN SIMULTANEOUSLY BE CONNECTED TO ONE OR
MORE DEVICES.

EXAMPLE

FG(1>3)&FG(7>9,,,2)>GLIS(220,440,200,2)ENV(50,100,500)
GLIS(440,880,1000)ENV(100,90,1200,1,10)

FG(12)&CHA(1)>ENV(50,100,1000)>GLIS(100,1000,1000)
IN THE LAST EXAMPLE THE GLIS TERM WILL HAVE NO EFFECT ON
THE CHA TERM.

FG(1>24)>RM(1,A,100);

IN THIS CASE ONLY CERTAIN FG:S CAN BE CONNECTED TO RM(1,A). THE ERROR PRI
WILL BE "ILLEGAL (DIS-)CONNECTION" BUT THE LEGAL CONNECTIONS ARE MADE BY
PROGRAM. *printout
by the*

FG(1)&FG(4,440)>FF(1)&CHA(2)
FG(7,220,100,1)>CHA(4,100)

THE ERROR IN THE FIRST LINE WILL NOT BE DETECTED BY THE PROGRAM UNTIL THE

FIRST SYMBOL ON THE SECOND LINE IS READ. THE SYMBOL IS NOT A & AND THUS
DEVICE CHAIN NR 2 ON THE FIRST LINE IS ENDED AND THE PROGRAM STARTS TO
CONNECT THE TWO DEVICE CHAINS. ERRORS ARE DETECTED AND AS USUAL A SCAN
IS MADE WHICH IN THIS CASE STOPS AT THE). ERROR PRINTOUT:

```
FG(      )  
02 051  ILLEGAL (DIS-)CONNECTION
```

IT WOULD HAVE BEEN BETTER TO END THE FIRST LINE WITH A ; WHICH SIGNALS ^{END} E
OF DEVICE CHAIN AND THE ERROR PRINTOUT WOULD HAVE COME AFTER THE FIRST L:

.EJECT

LINE

6. EMS1 COMMANDS

CONTENTS:

- 6.1 ABBREVIATIONS
- 6.2 BASIC SORT RECORD FILE OPERATIONS
- 6.3 SORT RECORD FILE COMMANDS
- 6.4 TEXT RECORD FILE COMMANDS
- 6.5 OTHER COMMANDS
- 6.6 TEMPORARY COMMANDS

6.1 ABBREVIATIONS

TEMP = FILE FOR STORING THE SORT-RECORDS OF THE TEMPORARILY STUDIED SOUND OBJECT

TEXT = FILE FOR STORING THE TEXT OF DITTO

ACC = FILE FOR STORING THE SORT-RECORDS OF THE CURRENT BLOCK (ACCUMULATED SOUND OBJECTS)

TACC = OUTPUT MEDIUM FOR STORING THE TEXT OF DITTO

AUX = AUXILIARY FILE FOR STORING A TEMPORARY MERGE OF TEMP AND ACC

MT = DIGITAL MAGNETIC TAPE

6.2 BASIC SORT RECORD FILE OPERATIONS

| OPERATION (NOT COMMAND) | DESCRIPTION |
|----------------------------|--|
| CLEAR | THE FILE IS CLEARED, I. E. THE FILE POINTER IS RESET TO THE BEGINNING OF THE FILE AND THE OLD POINTER IS SAVED FOR USE IN KEEP |
| KEEP | THE FILE POINTER IS RESTORED TO THE VALUE IT HAD BEFORE THE LAST CLEAR OPERATION ON THAT FILE |
| SORT | THE RECORDS IN THE FILE ARE SORTED ACCORDING TO TIME AND RECORD TYPE |
| MERGE | TWO SORTED FILES ARE MERGED INTO ONE FILE |
| RECORD | THE SORT-RECORDS ARE TRANSLATED TO EMS1-RECORDS AND RECORDED ON MT STARTING FROM THE CURRENT MT BLOCK POINTER (*) |
| PLAY | AT LEAST ONE MT BLOCK IS PLAYED IN THE STUDIO (*) |

(*) THE MT BLOCK POINTER IS LEFT IN THE NEW POSITION

6.3 SORT RECORD FILE COMMANDS

GLOBAL TIME IS COUNTED FROM THE START OF THE FIRST BLOCK.
BLOCK TIME IS COUNTED FROM THE START OF THE CURRENT BLOCK.
LOCAL TIME IS COUNTED FROM THE START OF THE CURRENT OBJECT (IN TEMP).

THE FORMAT FOR ALL THE TIME ENTRIES IS <SEC> [,<MS>].

THE LAST ENVELOPE (OR GLISSANDO) IN THE SORTED TEMP AND ACC STOPS
AT THE BLOCK TIMES <END> AND THE GLOBAL TIME <BLOCKEND> RESPECTIVELY.

<TERM=PEST> IS AN ESTIMATE OF THE NUMBER OF TERMS THAT MAY BE
ADDED TO THE CURRENT BLOCK.

| COMMAND | OPERATIONS |
|---------------------------------------|---|
| CLEAR | 1. TEMP IS CLEARED 2. TEXT IS CLEARED |
| CLEAR (MIX) | 1. ACC IS CLEARED 2. TACC IS ERASED (ONLY IF FILE,SEE BELOW) |
| CLEAR (ALL) = CLEAR & CLEAR (MIX) | |
| KEEP | 1. TEMP IS KEPT 2. TEXT IS KEPT |
| KEEP (MIX) | ACC IS KEPT |
| KEEP (ALL) = KEEP & KEEP (MIX) | |
| MIX [(<START>)] | 1. TEMP IS SORTED 2. TEMP AND ACC ARE MERGED INTO ACC (VIA AUX) ALL ENTRIES FROM TEMP ARE SORTED ACCORDING TO THE BLOCK TIME <START>+<LOCAL TIME>. IF <START> IS NOT GIVEN, <START>=0 IS ASSUMED. 3. TEMP IS CLEARED 4. TEXT IS ACCUMULATED TO TACC 5. TEXT IS CLEARED |
| APP | APPEND OBJECT. EQUIVALENT TO MIX (<END OF PRECEDING OBJECT>) |
| END | 1. ACC IS RECORDED AS A BLOCK STARTING FROM THE END OF THE PRECEDING BLOCK (<BLOCKEND>). 2. ACC AND TEMP ARE CLEARED 3. TEXT IS CLEARED 4. IF TACC IS A FILE: THE FILE <LABEL> SRC IS CLOSED. WARNING IF 56 BLOCKS ARE WRITTEN. |
| PLAY | 1. TEMP IS SORTED 2. TEMP IS RECORDED 3. TEMP IS PLAYED |

PLAY (MIX) 1. ACC IS RECORDED
 2. ACC IS PLAYED

PLAY (<LABEL1> [,<LABEL2>])
 1. THE BLOCK <LABEL1> IS PLAYED.
 [2. THE BLOCKS TO AND INCLUSIVE THE BLOCK
 <LABEL2> ARE PLAYED.]

PLAY (ALL) ALL THE RECORDED BLOCKS ARE PLAYED
 INCLUSIVE THE CURRENT BLOCK

TRY [(<START>)] 1. TEMP IS SORTED
 2. TEMP AND ACC ARE MERGED INTO AUX
 ALL ENTRIES FROM TEMP ARE SORTED ACCORDING
 TO THE BLOCK TIME <START>+<LOCAL TIME>.
 IF <START> IS NOT GIVEN, <START>=0 IS ASSUMED.
 3. AUX IS RECORDED
 4. AUX IS PLAYED

TRAPP TRY TO APPEND AN OBJECT EQUIVALENT TO
 TRY (<END OF PRECEDING OBJECT>).

WHEN THE COMMANDS MIX,APP,TRAPP OR TRY ARE GIVEN, THE MESSAGE
 ' <OBJECT NUMBER>, <START>, <END>, <TERM-REST> '

IS TRANSMITTED TO UNITS 3 AND 4, WHERE <OBJECT NUMBER> IS
 THE OBJECT NUMBER WITHIN THE BLOCK IN ORDER OF APPEARANCE AND
 <START> AND <END> ARE BLOCK TIMES.

WHEN THE COMMANDS END OR PLAY (MIX) ARE GIVEN, THE MESSAGE
 ' <LABEL>, <BLOCKSTART>, <BLOCKEND>, <TERM-REST> '

IS TRANSMITTED TO UNITS 3 AND 4, WHERE <BLOCKSTART> AND <BLOCKEND>
 ARE GLOBAL TIMES.

6.4 TEXT RECORD FILE COMMANDS

THE METASYMBOL <STRING> DENOTES A STRING VARIABLE. N DENOTES
 AN INTEGER. IF <EXT> IS NOT GIVEN BELOW, THE EXTENSION
 'SRC' IS ASSUMED.

| COMMAND | OPERATIONS |
|-----------------------------|---|
| TOP (<FNAME> [, <EXT>]) | SET LINE POINTER IN FILE <FNAME> <EXT> TO THE FIRST LINE OF THE FILE. USE IN AND SKIP TERMS TO GET LINES FROM THE FILE. |
| IN [(N)] | READ N LINES FROM THE FILE WHOSE NAME WAS GIVEN IN THE PREVIOUS TOP TERM. THE LINE POINTER IS MOVED N LINES FORWARD IN THE FILE. IF N IS NOT GIVEN, N=1 IS ASSUMED |

SKIP [(N)] MOVE THE LINE POINTER N LINES FORWARD IN THE FILE WHOSE NAME WAS GIVEN IN THE PREVIOUS TOP TERM. IF N IS NOT GIVEN, N=1 IS ASSUMED

LOOK [(N)] READ N LINES FROM THE FILE WHOSE NAME WAS GIVEN IN THE PREVIOUS TOP TERM AND DISPLAY THEM ON UNIT 4 WITH LINE NUMBERS. THE LINE POINTER IS NOT MOVED. IF N IS NOT GIVEN, N=1 IS ASSUMED.

CALL (<FNAME> [,<EXT>]) ADD THE CONTENTS OF THE FILE <FNAME> <EXT> TO THE CURRENT EMS1-TEXT IN TEXT (EQUIVALENT TO 'TOP(<FNAME>)IN(100000)')

SAVE (<FNAME> [,<EXT>]) SAVE THE CURRENT TEXT IN A NEW FILE <FNAME> <EXT>.

REPL (<FNAME> [,<EXT>]) REPLACE THE CONTENTS OF THE OLD FILE <FNAME> <EXT> WITH THE CURRENT TEXT.

ERASE (<FNAME> [,<EXT>]) ERASE THE FILE <FNAME> <EXT>.

6.5 OTHER COMMANDS

| COMMAND | OPERATIONS |
|-------------------------------|--|
| BEGIN (<LABEL>) | STORE THE BLOCK NAME <LABEL> ON MT FOR LATER USE WITH THE PLAY COMMAND. IF TACC IS A FILE: OPEN <LABEL> SRC. |
| DELETE (<S1>,<S2>,...,<SN>) | DELETE SYMBOLS <S1>,<S2>,...,<SN>. THE SYMBOLS MAY BE VARIABLES OR MACROS. |
| EXIT | EXIT AND RESTART THE PROGRAM |
| LIST | LIST EXPANDED MACROS |
| MESS (<TEXT>) | <TEXT> WILL BE WRITTEN ON UNIT 4. EMS1 SWITCHES TO ERROR MODE |
| MEX | EXIT OUT OF MACRO TO LEVEL 0. |
| NOLIST | NO LISTING OF MACRO EXPANSIONS |
| STDTIM (<TIME CONSTANT>) | THE TIME SCALE IS CHANGED TO 1000/<TIME CONSTANT> OF THE MUSIC TIME SCALE. |
| WRITE (<S1>,<S2>,...,<SN>) | <S1>,<S2>,...,<SN> ARE DEFINED OR UNDEFINED SYMBOLS. THE VALUES OF THESE SYMBOLS WILL BE WRITTEN ON UNITS 3 AND 4. EXAMPLES OF PRINTOUTS (<S1>='MIN',<S2>='LIMIT',<S3>='M7'): |
| | 'MIN = 15 TYPE: 01' |
| | 'LIMIT NOT DEFINED ' |
| | 'M7 *MACRO* ' |

6.6 TEMPORARY COMMANDS

THE TEMPORARY COMMANDS

CALL, IN, LOOK, PLAY, SAVE, SKIP, TOP, TRY, TRAPP

ARE EXECUTED BUT NEVER STORED IN THE TEMPORARY TEXT FILE 'TEXT' AND IN THE ACCUMULATED TEXT MEDIUM 'TACC' .

CALL, ERASE, IN, LOOK, REPL, SAVE, SKIP AND TOP ARE IGNORED WHEN READ FROM A SECONDARY INPUT FILE.

.EJECT

7. PSEUDOOPERATIONS

PSEUDOOPERATIONS ARE SYMBOLS WHICH CONTROL THE COMPILATION OF THE INPUT TEXT.

A CONDITIONAL EXPRESSION IS WRITTEN :

-CONDITIONAL SYMBOL-(=SYMBOL TO BE TESTED-) - TEXT - <
THE TEXT MAY CONTINUE ON MORE THAN ONE LINE. THE < IS ENDMARK OF THE CONDITIONAL. IF THE CONDITION IS NOT SATISFIED THE TEXT AFTER THE CONDITIONAL IS TREATED LIKE A COMMENT UNTIL THE ENDMARK. THE CONDITIONALS IMPLEMENTED IN THE SYSTEM ARE:

| CONDITIONAL SYMBOL | CONDITION TO BE TESTED |
|--------------------|------------------------|
|--------------------|------------------------|

| | |
|-------|---------------------|
| IFUND | IF UNDEFINED |
| IFDEF | IF DEFINED |
| IFPOS | IF POSITIVE (.GT.0) |
| IFZER | IF ZERO (.EQ.0) |
| IFNEG | IF NEGATIVE (.LT.0) |

EX. INPUT TEXT: IFUND(NR)NR=0;< NR=NR+1
THIS MEANS THAT IF THE SYMBOL 'NR' IS UNDEFINED 'NR' WILL BE ASSIGNED THE VALUE 0 AND AFTER THAT 1 IS ADDED TO 'NR'.
IF 'NR' IS DEFINED BEFORE THE TEXT: NR=0; WILL NOT BE COMPILED I.E. 'NR' WILL KEEP ITS PREVIOUS VALUE. AFTER THAT 1 IS ADDED TO 'NR'.

7.1 NESTED CONDITIONALS

THE TEXT CORRESPONDING TO A CONDITIONAL MAY CONTAIN CONDITIONALS. THE INNER CONDITIONALS WILL HAVE THE SAME ENDMARK AS THE OUTERMOST.

EX. IFDEF(IND)FG(1,220,80,1)>CHA(1,100)IFZER(FR)FR=1000;<
POSSIBLE INTERPRETATIONS OF THIS LINE, DEPENDING ON THE PREVIOUS ASSIGNMENTS OF 'IND' AND 'FR' ARE:

'IND' DEFINED, 'FR' ZERO: FG(1,220,80,1)>CHA(1,100)FR=1000;
'IND' DEFINED, 'FR' NONZERO: FG(1,220,80,1)>CHA(1,100)
'IND' DEFINED, 'FR' UNDEFINED: IFZER(FR)
04 001 SYMBOL NOT DEFINED
'IND' UNDEFINED, 'FR' IRRELEVANT: NO RESULTING OUTPUT

.EJECT

8. MACRO

FREQUENTLY CERTAIN SECTIONS OF TEXT WILL BE REPEATED SEVERAL TIMES IN THE INPUT, USUALLY WITH ONLY SOME MINOR MODIFICATIONS. IN ORDER TO AVOID REPEATING THE TEXT STRING MORE THAN ONCE, IT MAY BE ASSIGNED TO A SYMBOL AS IN CHAPTER 2.6 SUBSEQUENTLY, ANY OCCURRENCE OF THIS MACRO SYMBOL FOLLOWED BY ANY OTHER DELIMITER THAN '=' WILL CAUSE THE SYMBOL TO BE REPLACED BY THE TEXT STRING.

THE NUMERICAL VALUES OF VARIABLES INSIDE THE MACRO WILL BE SUBSTITUTED AFTER THE DEFINING STRING HAS BEEN COPIED INTO THE TEXT IN PLACE OF THE MACRO SYMBOL. IT IS NOT NECESSARY TO HAVE VARIABLES OCCURRING INSIDE A MACRO DEFINED WHEN THE TEXT STRING IS ASSIGNED TO THE MACRO SYMBOL. BY GIVING DIFFERENT VALUES TO SUCH VARIABLES EACH TIME BEFORE THE MACRO IS CALLED FOR IN THE INPUT TEXT, DIFFERENT PARAMETERS WILL APPEAR INSIDE THE CORRESPONDING TEXTSTRINGS. ALTERNATIVELY, THE MACRO DEFINITION ITSELF MAY BE USED TO CHANGE VARIABLES FROM THEIR INITIAL VALUES.

HOW TO DEFINE A MACRO IS DISCUSSED EARLIER IN CHAPTER 2.6 THE VARIABLES IN A MACRO MAY BE CHANGED OUTSIDE THE MACRO.

EX. SET THE FIRST 3 SOUNDGENERATORS TO FREQUENCIES 100,200,300 INTENSITIES TO 100,90,80 AND WAVEFORMS TO 2,4,6

```
SETFG="FG(NR,FR,NI,VA)NR=NR+1;FR=FR+100;NI=NI-10;VA=VA+2;"
NR=1;FR=100;NI=100;VA=2
```

```
SETSG;SETSG;SETSG
```

THE RESULT FROM THE LAST LINE IS THE SAME AS WRITING:

```
FG(1,100,100,2) FG(2,200,90,4) FG(3,300,80,6)
```

BECAUSE THE VARIABLES 'NR', 'FR', 'NI', 'VA' ARE CHANGED INSIDE

THE MACRO EACH TIME THE MACRO IS CALLED, I.E. BY WRITING 'SETSG'.

THE SAME RESULT IS OBTAINED FROM THE FOLLOWING TEXT:

```
SETSG="IFUND(NR)NR=1;FR=100;NI=100;VA=2;<FG(NR,FR,NI,VA)
```

```
NR=NR+1;FR=FR+100;NI=NI-10;VA=VA+2;"
```

```
SETSG;SETSG;SETSG
```

8.1 NESTING OF MACROS

A MACROTEXT MAY CONTAIN CALLS TO OTHER MACROS. MACROS CALLED FOR IN THE INPUT TEXT ARE CALLED FIRST LEVEL MACROS, MACROS CALLED IN A FIRST LEVEL MACRO IS CALLED A SECOND LEVEL MACRO, A MACRO CALLED IN A SECOND LEVEL MACRO IS CALLED A THIRD LEVEL MACRO ETC.

THE MAXIMUM AMOUNT OF LEVELS MAY BE 100 AT PRESENT.

EX. WE WANT TO PUT AN ENVELOPE ON OUTPUT CHANNEL 1 WHICH GOES FROM OLD-5 TO OLD+5 IN 500 MS AND THEN FROM OLD+5 TO OLD-5 IN ANOTHER 500 MS. WE WANT THIS REPEATED 10 TIMES.

```
EA1="ENV(OLD-5,OLD+5,500,3)>ENV(OLD+5,OLD-5,500,3)"
```

```
EA5="EA1>EA1>EA1>EA1>EA1"
```

```
EA10="EA5>EA5"
```

```
EAMP(CH1)>EA10
```

8.2 RECURSIVE CALLS

A MACRO MAY CONTAIN A CALL FOR ITSELF BUT IT IS NECESSARY TO USE CONDITIONALS TO STOP THE RECURSION OTHERWISE IT WOULD BE AN INFINITE

PROCESS.

EX. THE SAME RESULT AS IN PARAGRAPH 8.0 COULD BE OBTAINED IN THE FOLLOWING WAY:

```
SETSG="IFUND(NR)NR=1;FR=100;NI=100;VA=2;<TEST=4-NR;IFPOS(TEST)
SG(NR,FR,NI,VA)NR=NR+1;FR=FR;100;NI=NI-10;VA=VA+2;SETSG;<"
```

8.3 MACRO EXPANSION

IF IN THE LIST MODE THE WHOLE MACRO-TEXT IS TRANSFERRED TO THE OUTPUT TEXT.

EX.

INPUT TEXT

```
MAC1="ENV(0,100,500)"
MAC2="MAC1>MAC1>"
MAC3="MAC2>MAC1>ENV(100,0,500)"
MAC3
```

OUTPUT TEXT

```
'00101' MAC1="ENV(0,100,500)"
'00102' MAC2="MAC1>MAC1>"
'00103' MAC3="MAC2>MAC1>ENV(100,0,500)"
'00104' '---MACRO---: MAC3'
'00105' '---MACRO---: MAC2'
'00106' '---MACRO---: MAC1'
'00107' ENV(0,100,500)
'00108' >
'00109' '---MACRO---: MAC1'
'00110' ENV(0,100,500)
'00111' >
'00112' >
'00113' '---MACRO---: MAC1'
'00114' ENV(0,100,500)
'00115' >ENV(100,0,500)
```

A LATER CORRECTIVE RUN, USING THIS OUTPUT TEXT AS INPUT, CAN BE USED TO CORRECT THE RESULT OF THE MACRO EXPANSION. AFTER THE 'NDLIST' COMMAND, THE EXPANSION OF MACROS IN THE OUTPUT TEXT IS SUPPRESSED, AND CORRECTIONS CAN THEN ONLY BE MADE AT THE MACRO LEVEL.

.EJECT

9. OPERATING PROCEDURES

IF THE COMPUTER IS RUNNING AND THE MONITOR SYSTEM LOADED
START AT STEP 6 ELSE DO THE FOLLOWING

- 1 TURN ON THE POWER SWITCHES ON THE COMPUTER AND
THE CONSOLE TELETYPE
- 2 PLACE THE PAPER TAPE MARKED 'DECDISK BOOTSTRAP' IN THE
PAPER TAPE READER
- 3 SET THE ADDRESS SWITCHES (THE UPPER SWITCH ROW WITH
15 SWITCHES) TO '77637' (111 111 110 011 111)
- 4 PRESS STOP AND RESET
- 5 PRESS READIN

THE MONITOR WILL NOW BE LOADED AND STARTED.
IT WILL IDENTIFY ITSELF BY PRINTING

```
KM15 VXX          (XX IS VERSION NUMBER)
$
```

ON THE CONSOLE TELETYPE

- 6 COPY THE EMS1-TAPE TO DISK IN THE FOLLOWING WAY

MOUNT THE EMS1 DECTAPE ON A DECTAPE TRANSPORT AND
SELECT NUMBER 1 ON THAT TRANSPORT (BE CAREFULLY SO THAT
NO OTHER TRANSPORT IS READY AT THE SAME TIME WITH THE
SAME NUMBER)

CALL THE SYSTEM PROGRAM PIP

```
SPIP          ($ TYPED BY MONITOR, PIP BY USER)
```

```
PIP VXX      (PIP IS LOADED AND READY TO USE)
>C DK5 0 DT1 (H)
```

```
>^C          (> TYPED BY PIP, THE REST OF THE LINE BY USER)
              (PIP DONE KILL PIP)
              (^C MEANS HOLD DOWN CNTRL KEY WHILE
              STRIKING C)
```

- 7 ASSIGN THE UNITS USED BY EMS1 PACKAGE TO THE MEDIA
YOU WANT TO USE IN THIS PARTICULAR RUN.
THE UNITS USED ARE THE FOLLOWING

| UNIT | USE | RECOMMENDATION |
|------|----------------|----------------|
| -14 | SORT DISK | DK4 |
| -15 | SORT DISK | DK5 |
| -10 | SORT DISK | DK6 |
| -12 | ERROR MESSAGES | TT OR TV |
| -4 | EXECUTE FILE | DK1 |

| | | |
|----|--------------------------|----------|
| 1 | STANDARD SYMBOL DK1 | |
| | INPUT AND | |
| | INTERMEDIATE STORAGE | |
| 2 | TEXT INPUT | DT1 |
| 3 | TEXT OUTPUT | DT2 |
| | "TEXT" (MUST BE FILEOR.) | |
| 4 | INPUT OF | TT OR TV |
| | CORRECTIONS (NONFILEOR.) | |
| 5 | TEXT OUTPUT | DT3 |
| | "TACC" | |
| 6 | SAVE TEXT | DT1 |
| 7 | ACCUMULATED | MT0 |
| | EMS CODE | |
| 10 | ACCUMULATED | MT0 |
| | EMS CODE | |

EXAMPLE

\$A DK4 -14/DK5 -15/DK6 -10/DT1 3/DT2 5/DT3 2,6/MT1 7,10
 \$A DK7 -4,1/TT 4,-12

LOAD AND START THE EMS1 PACKAGE BY TYPING
 AFTER THE MONITORS '\$' SIGN

E EMS1

WHEN EMS1 IS LOADED AND STARTED IT TYPES

EMS1 VXX (XX VERSION NUMBER)

ON THE UNIT ASSIGNED TO 4
 IF UNIT 2 FILEORIENTED IT TYPES:

NAME OF INPUT FILE?

ANSWER WITH FILENAME, EXT OF MAIN TEXT INPUT FILE OR WITH A
 CARRIAGE RETURN IF NO MAIN FILE EXISTS. IN THE LATTER CASE
 MAIN TEXT INPUT IS UNIT 4 BUT SECONDARY TEXT INPUT IS STILL UNIT 2.
 ANSWER WITH ALT. MODE IF END OF RUN. EMS1 RETURNS TO MONITOR.

IF UNIT 2 FILEOR. OR NOT THE NEXT QUESTION IS:

INTERACTIVE MODE. YES OR NO?

ANSWER 'NO' IF NO ERROR CORRECTIONS ARE TO BE MADE. ERROR PRINTOUTS
 WILL BE AS USUAL.

ANSWER YES IF CORRECTIONS ARE TO BE MADE FROM UNIT 4 IF AN ERROR
 IS DETECTED BY THE SYSTEM.

THE PROGRAM IS NOW READY TO USE.

.EJECT

APPENDIX 1

PERMANENT SYMBOLS

DEVICE SYMBOLS

FG
NG
FF
REV
RM
AM
AMP
CD

ENVELOPE AND GLISSANDI TERMS

ENV
T
Z
ESTEP
GLIS
GSTEP

COMMANDS

PLAY
TRY
TRAPP
APP
MIX
END
STDTIM
CLEAR
KEEP
CALL
TOP
IN
SKIP
SAVE
ERASE
REPL
DELETE
WRITE
MESS
MEX
LOOK
EXIT

CONDITIONALS

IFDEF
IFUND
IFPOS
IFZER
IFNEG

AUXILIARY SYMBOLS

LIST
NOLIST
OLD

ARGUMENTS TO CLEAR AND KEEP COMMANDS

MIX
ALL

ARGUMENTS TO RM AND AM TERMS

A
B

ARGUMENTS TO NG TERM

WHITE
PINK

SPECIAL CONNECTION POINTS

FG3 FREQ. GENERATORS GROUP 1-3
FG6 FREQ. GENERATORS GROUP 4-6
FG9 FREQ. GENERATORS GROUP 7-9
FG12 FREQ. GENERATORS GROUP 10-12
FG15 FREQ. GENERATORS GROUP 13-15
FG18 FREQ. GENERATORS GROUP 16-18
FG21 FREQ. GENERATOR GROUP 19-21
FG24 FREQ. GENERATOR GROUP 22-24
FS FREQUENCY SHIFTER

.EJECT

APPENDIX 2. LEGAL STUDIO CONNECTIONS.

| FROM | TO |
|---------|-----------|
| FG (1) | CHA (1) |
| FG (2) | CHA (2) |
| FG (3) | CHA (3) |
| FG3 | CHA (4) |
| | CD (1) |
| | FG6 |
| FG (4) | CHA (1) |
| FG (5) | CHA (2) |
| FG (6) | CHA (3) |
| FG6 | CHA (4) |
| | CD (1) |
| | FG9 |
| | RM (1, A) |
| | RM (1, B) |
| | RM (2, B) |
| | REV (1) |
| | REV (2) |
| | AM (1, B) |
| | FF (1) |
| | FF (2) |
| FG (7) | CHA (1) |
| FG (8) | CHA (2) |
| FG (9) | CHA (3) |
| FG9 | CHA (4) |
| | CD (1) |
| | FG12 |
| FG (10) | CHA (1) |
| FG (11) | CHA (2) |
| FG (12) | CHA (3) |
| FG12 | CHA (4) |
| | CD (1) |
| | FG15 |
| | RM (1, A) |
| | RM (1, B) |
| | RM (2, B) |
| | REV (1) |
| | REV (2) |
| | AM (1, B) |
| | FF (1) |
| | FF (2) |
| FG (13) | CHA (1) |
| FG (14) | CHA (2) |
| FG (15) | CHA (3) |
| FG15 | CHA (4) |
| | CD (1) |
| | FG18 |

.EJECT

| FROM | TO |
|--------|---------|
| FG(16) | CHA(1) |
| FG(17) | CHA(2) |
| FG(18) | CHA(3) |
| FG18 | CHA(4) |
| | CD(1) |
| | FG21 |
| | RM(1,B) |
| | RM(2,B) |
| | REV(1) |
| | REV(2) |
| | AM(1,B) |
| | AM(2,B) |
| | FF(1) |
| | FF(2) |
| FG(19) | CHA(1) |
| FG21 | CHA(2) |
| | CHA(3) |
| | CHA(4) |
| | CD(1) |
| | FG24 |
| FG(20) | FG21 |
| | FS |
| FG(21) | FG21 |
| | RM(1,A) |
| FG24 | CHA(1) |
| | CHA(2) |
| | CHA(3) |
| | CHA(4) |
| | CD(1) |
| | RM(1,B) |
| | RM(2,B) |
| | REV(1) |
| | REV(2) |
| | AM(1,B) |
| | AM(2,B) |
| | FF(1) |
| | FF(2) |
| FG(22) | FG24 |
| | RM(2,A) |
| FG(23) | FG24 |
| | AM(1,A) |
| FG(24) | FG24 |
| | AM(2,A) |

.EJECT

| FROM | TO |
|--------|---|
| NG | CHA(1) CHA(2) CHA(3) CHA(4) CD(1) RM(2,B) REV(1) AM(2,B) FF(1) FF(2) |
| FF(1) | CHA(2) CHA(3) CHA(4) CD(1) RM(1,A) RM(1,B) RM(2,B) REV(2) AM(1,A) AM(2,B) FF(2) AMP(1) AMP(2) |
| FF(2) | CHA(2) CHA(3) CHA(4) CD(1) RM(2,B) REV(2) AM(1,B) AM(2,A) FF(1) AMP(1) AMP(2) |
| REV(1) | CHA(1) CHA(2) CHA(3) CHA(4) CD(1) AM(1,B) AM(2,B) AMP(1) AMP(2) |

.EJECT

| FROM | TO |
|--------|---|
| REV(2) | CHA(1) CHA(2) CHA(3) CHA(4) CD(1) AM(1,B) AM(2,B) AMP(1) AMP(2) |
| RM(1) | CHA(1) CHA(2) CHA(3) CHA(4) CD(1) REV(2) AM(1,A) AM(2,B) FF(1) FF(2) AMP(1) AMP(2) |
| RM(2) | CHA(1) CHA(2) CHA(3) CHA(4) CD(1) REV(2) AM(1,B) AM(2,A) FF(1) FF(2) AMP(1) AMP(2) |
| RM(3) | CHA(1) CHA(2) CHA(3) CHA(4) CD(1) REV(2) AM(2,B) FF(1) FF(2) AMP(1) AMP(2) |
| AM(1) | CHA(1) CHA(2) CHA(3) CHA(4) CD(1) |

.EJECT

| FROM | TO |
|--------|--|
| AM(2) | CHA(1) CHA(2) CHA(3) CHA(4) CD(1) |
| AMP(1) | RM(1,B) RM(2,B) REV(2) AM(1,B) AM(2,B) FF(1) FF(2) |
| AMP(2) | RM(1,B) RM(2,B) REV(2) AM(1,B) AM(2,B) FF(1) FF(2) |

.EJECT

APPENDIX 3. THE SORT RECORD

| WORD | TYPE OF SORTRECORD | CONTENTS |
|------|--------------------|-------------------------------------|
| 1 | ALL TYPES | LOCAL TIME |
| 2 | ALL TYPES | BITS 0-5: LOCAL TIME (CONT) |
| | | BITS 6-17: STUDIO ADDRESS |
| 3 | ALL TYPES | BITS 0-11: SOURCE LINE NR |
| | | BITS 12-17: TERM NR |
| 4 | CONNECTIONS | =0 IF DISCONNECT, =1 IF CONNECT |
| 4 | AMPLIFIERS | AMPLITUDE IN DB*4 |
| 4 | FG FREQUENCY | FREQUENCY IN HZ |
| 4 | FG WAVEFORM | WAVEFORM |
| 4 | ENV, GLIS | FROM AMPLITUDE(FREQUENCY) |
| 5 | NOT ENV, GLIS | SAME AS WORD 4 |
| 5 | ENV, GLIS | TO AMPLITUDE(FREQUENCY) |
| 6 | NOT ENV, GLIS | BIT 17=1 IF OLD+VALUE |
| 6 | ENV, GLIS | BITS 0-15 ENVELOPE TYPE (GLIS TYPE) |
| | | BIT 16=1 IF OLD+DATA 2 |
| | | BIT 17=1 IF OLD+DATA 1 |
| 7 | NOT ENV, GLIS | NOT USED |
| 7 | ENV, GLIS | ENVELOPE DURATION |
| 8 | NOT ENV, GLIS | NOT USED |
| 8 | ENV, GLIS | ENVELOPE (GLIS) STEP |

.EJECT

APPENDIX 4. ERROR PRINTOUTS.

| | | |
|----|-----|---|
| 02 | 011 | ILLEGAL DELIMITER |
| 02 | 012 | ILLEGAL PARAMETER |
| 02 | 013 | ILLEGAL MNEMONIC |
| 02 | 014 | ILLEGAL NUMBER OF PARAMETERS |
| 02 | 021 | ILLEGAL FREQUENCY GENERATOR NUMBER |
| 02 | 022 | ILLEGAL FREQUENCY GENERATOR FREQUENCY |
| 02 | 023 | ILLEGAL FREQUENCY GENERATOR WAVEFORM |
| 02 | 024 | ILLEGAL FREQUENCY GENERATOR INTENSITY |
| 02 | 032 | ILLEGAL FREQUENCY FILTER CHANNEL |
| 02 | 033 | ILLEGAL FREQUENCY FILTER INTENSITY |
| 02 | 041 | ILLEGAL AMPLIFIER NUMBER |
| 02 | 042 | ILLEGAL AMPLIFIER INTENSITY |
| 02 | 051 | ILLEGAL (DIS-) CONNECTION |
| 02 | 062 | ILLEGAL REVERBATION TIME |
| 02 | 063 | ILLEGAL REVERBATION INTENSITY |
| 02 | 070 | TIME MISSING IN T-TERM |
| 02 | 071 | ILLEGAL TIME |
| 02 | 081 | ILLEGAL NOISE COLOUR |
| 02 | 082 | ILLEGAL NOISE INTENSITY |
| 02 | 091 | ILLEGAL AMPLITUDE MODULATOR NUMBER |
| 02 | 092 | ILLEGAL AMPLITUDE MODULATOR ENTRY |
| 02 | 093 | ILLEGAL AMPLITUDE MODULATOR INTENSITY |
| 02 | 102 | ILLEGAL RING MODULATOR ENTRY |
| 02 | 103 | ILLEGAL RING MODULATOR INTENSITY |
| 02 | 112 | ILLEGAL CHANNEL DISTRIBUTOR INTENSITY |
| 02 | 121 | ILLEGAL DEVICE NUMBER |
| 02 | 150 | ILLEGAL ANALOG TAPE INTENSITY |
| 02 | 142 | ILLEGAL CHANNEL INTENSITY |
| 02 | 201 | ILLEGAL ESTEP VALUE |
| 02 | 241 | ILLEGAL ENV SYNTAX |
| 02 | 242 | ILLEGAL ENV AMPLITUDE |
| 02 | 245 | ENV OR GLISTIME LESS THAN STEP |
| 02 | 246 | ILLEGAL ENV TIME |
| 02 | 247 | ILLEGAL ENV TYPE |
| 02 | 251 | NO DEVICE TO ENVELOPE GIVEN |
| 02 | 261 | ILLEGAL LOCAL TIME VALUE |
| 02 | 271 | ILLEGAL GSTEP VALUE |
| 02 | 301 | NO DEVICE TO GLISSANDO GIVEN |
| 02 | 302 | ILLEGAL GLIS SYNTAX |
| 02 | 303 | ILLEGAL GLIS TIME |
| 02 | 304 | ILLEGAL GLIS TYPE |
| 02 | 305 | ILLEGAL GLIS FREQUENCY |
| 02 | 142 | ILLEGAL CHANNEL INTENSITY |
| 04 | 001 | SYMBOL WITH MORE THAN 6 CHARACTERS. |
| 04 | 002 | INTEGER WITH MORE THAN 5 FIGURES |
| 04 | 003 | TWO DECIMAL POINTS |
| 04 | 004 | A DECIMAL POINT (.) MUST NOT BE USED AS A DELIMITER. |
| 04 | 010 | IN THE EXPRESSION 'A=B' B IS NOT A DEFINED SYMBOL. |
| 04 | 020 | SYMBOL NOT DEFINED. |
| 04 | 021 | SYMBOL DEFINED BUT NOT POSSIBLE IN THIS PART OF THE TEXT. |
| 04 | 031 | TOP NOT EXECUTED IF READ FROM SECONDARY TEXT INPUT |
| 04 | 032 | UNIT 2 NOT FILEORIENTED |
| 04 | 033 | IN,SKIP,DELETE,LOOK NOT EXECUTED IF READ FROM SEC. TEXT INPUT |
| 04 | 034 | FILE NOT PRESENT ON UNIT 2 |
| 04 | 035 | NO TOP COMMAND EXECUTED OR SECONDARY FILE EMPTY |

```

04 036 PARAMETER NOT INTEGER
04 050 LABEL ERROR
04 064 : / ^ OR ) NOT POSSIBLE AS FIRST ELEMENT OF AN EXPR.
04 074 OVERFLOW. ADD OR SUBTRACT
04 075 OVERFLOW. MULTIPLICATION
04 076 TRY TO DIVIDE BY ZERO
04 100 A '=' IS POSSIBLE ONLY AFTER A SYMBOL OR A '!='.
04 101 ATTEMPT TO ASSIGN A FIXED SYMBOL
04 102 SYMBOL TABLE FULL
04 104 A ' " ' FOUND IN A NON-MACRO TEXT.
04 103 A MACRO MUST NOT BE CALLED IN ERROR MODE.
05 001 END OF INPUT FILE WITHOUT AN 'EXIT'
INPUT FROM UNIT 4.
09 001 NO EXTENSION TO FILENAME
09 002 INPUT FILE FILEORIENTED. FILENAME ?
09 003 OUTPUT FILE FILEORIENTED. FILENAME ?
09 004 FILE NOT FOUND.
09 005 UNIT 1 NOT FILEORIENTED.
09 006 FILE FMNEMO EMS NOT FOUND
09 007 FILE ERROR EMS NOT FOUND.
09 008 NO FILENAME ON FILEORIENTED DEVICE.
09 99 ----- EXIT. END OF THIS RUN.-----
01 001 NOT A PERMANENT SYMBOL
01 002 NOT A COMMAND
01 003 ILLEGAL PARAMETER IN 'PLAY' TERM
01 004 NOTHING TO PLAY
01 005 ILLEGAL PARAMETER IN 'TRY'-TERM
01 006 ILLEGAL PARAMETER IN 'MIX'-TERM
01 007 THE ONLY POSSIBLE PARAMETERS TO 'CLEAR' ARE 'MIX' OR 'ALL'
01 009 THE ONLY POSSIBLE PARAMETERS TO 'KEEP' ARE 'MIX' OR 'ALL'
01 010 KEEP WITHOUT MEANING HERE
01 015 SAVE,ERASE,REPL NOT EXECUTED IF READ FROM SEC. TEXT INPUT
01 016 LEFT PARANTHESIS MISSING AFTER SYMBOL
01 017 , AFTER SYMBOL
01 018 'END' AFTER PREVIOUS 'BEGIN' MISSING
01 020 BEGIN COMMAND FIRST THING IN A BLOCK!
01 021 'BEGIN' AFTER PREVIOUS 'END' MISSING
01 024 LEFT PARANTHESIS MISSING IN 'STDTIM'-TERM
01 025 ERRONEUS TIMEPARAMETER IN 'STDTIM'-TERM
01 030 FILE ALREDY PRESENT
01 040 ERROR DETECTED IN "MERGE". NOT ENOUGH SPACE ON OUTPUT UNIT
01 041 DURATION OF MIX =0!
01 050 LABEL ERROR
05 002 END OF FILE ON ADDITIONAL INPUT FILE.
09 002 EMS1 V1.1

09 004 NAME OF INPUT FILE?
09 005 UNIT 3 MUST BE FILEORIENTED.
09 006 INTERACTIVE MODE? YES OR NO?
09 007
00 000
.EJECT

```

APPENDIX 6. EXAMPLES OF EMS1 TEXT.

EXAMPLE NR 1.

```

'00001' BEGIN(STORS)
'00002' G0=196;A0=220;D0=147;H00=123
'00003' C1=262;D1=294;E1=330;F1=349;H0=247;A0=220;C0=131
'00004' F0=175;G0=196
'00005' NOT1=1000;NOT2=500;MAX=80;WF=2;ET1=-1;ET2=-2
'00006' TON="LT(GT1,GT2)MAX1=MAX+10;T1=TID/10;T2=TID-2*T1;T3=2*T1
'00007' GT2=GT2+TID;GT3=GT2-1000;IFPOS(GT3)GT2=GT3;GT1=GT1+1;<
'00008' FR1=FR+10
'00009' IFDEF(LISTA)WRITE(MAX1,T1,T2,T3,GT,FR1)MESS(???)<
'00010' FG(NR,,WF)>ENV(50,MAX1,T1,2)>ENV(MAX1,MAX,T1,ET2)>T(T2)
'00011' GLIS(FR1,FR,T3)
'00012' FR1=FR+2;NR1=NR+1;MAX1=MAX-6
'00013' FG(NR1,FR1,,WF)>ENV(50,MAX,T1)>ENV(MAX,40,T2)
'00014' FR1=FR1+2;NR1=NR1+1;MAX2=MAX-10
'00015' FG(NR1,FR1,,WF)>ENV(50,MAX1,T1,ET1)>ENV(MAX1,MAX2,T1)>ENV(MAX2,30,T2)"
'00016' SILL="TID=NOT1;FR=C1;TON
'00017' TID=NOT2;FR=D1;TON
'00018' FR=C1;TON
'00019' TID=NOT1;FR=H0;TON
'00020' FR=C1;TON
'00021' FR=E1;TON
'00022' TID=NOT2;FR=F1;TON
'00023' FR=E1;TON
'00024' TID=NOT1;FR=D1;TON
'00025' FR=E1;TON
'00026' FR=C1;TON
'00027' TID=NOT2;FR=A0;TON
'00028' FR=F0;TON
'00029' TID=NOT1;FR=G0;TON
'00030' FR=C0;TON
'00031' "
'00032' FG3>FG6>CHA(1,100)
'00033' FG9>FG12>CHA(2,100)
'00034' FG15>FG18>CHA(3,100)
'00035' FG(21)>FG21>CHA(4,100)
'00036' FG(20)>FG21
'00037' FG(24)>FG24>CHA(4,100)
'00038' FG(23)>FG24
'00039' FG(22)>FG24
'00040' GT1=1;GT2=0
'00041' NR=1;SILL;SILL
'00042' NR=7;GT1=5;GT2=0;SILL;SILL
'00043' NR=13;GT1=9;GT2=0;SILL;SILL
'00044' NR=19;GT1=13;GT2=0;SILL;SILL
'00045' MIX
'00046' END

```

DESCRIPTION OF THE EMS1-TEXT.

'SILL' IS A MACRO FOR A 15 NOTE CANON. FOUR DIFFERENT SOUNDGENERATOR GROUPS EACH PLAY THIS TUNE. THEY START TO SOUND AFTER RESP. 1,5,9,13 SECONDS. ONE NOTE IS BUILT BY THE MACRO 'TON'. THIS MACRO CONTAINS A GROUND NOTE FOR SOUNDGENERATOR NR "NR" AND 3 OVERTONES ON SOUNDGENERATORS "NR+1", "NR+2" AND "NR+3"

EACH OF THE TONES HAS SPECIFIC ENVELOPES TO GIVE THE WHOLE TONE A SPECIFIC CHARACTERISTIC. THE 'TON' MACRO STARTS WITH A CALCULATION OF DURATION TIMES OF THE ENVELOPES. IF THE SYMBOL 'LISTA' IS DEFINED THE CALCULATED VALUES OF THE SYMBOLS MAX1, T1, T2, T3, GT1 AND FR1 ARE WRITTEN OUT AND AFTER THAT THE MESSAGE " ??? " IS WRITTEN TO THE USER AND THE SYSTEM WAITS IN ERROR MODE. THIS GIVES THE USER AN OPPORTUNITY TO CHANGE VALUES. BY A "DELETE(LISTA)" THIS MESSAGE WON'T BE WRITTEN OUT UNTIL 'LISTA' IS DEFINED AGAIN. THE CONNECTIONS ARE MADE SO THAT SOUNDGEN. 1-6 IS CONNECTED TO CHANNEL 1, 7-12 TO CHANNEL 2, 13-18 TO CHANNEL 3 AND 19-24 TO CHANNEL 4. THE 'SILL' MACRO "PLAYS" THE CANON ONCE THUS EACH GROUP "PLAYS" THE TUNE TWICE. THE 'MIX' COMMAND TRANSFERS THE 'TEMP' DISK TO THE 'MIX' DISK. THE 'END' COMMAND IMPLIES GENERATION OF MT-CODE ON THE MT. THE BL LABEL ON THE MT WILL BE 'STORS'. THIS LABEL CAN BE USED IN A LATER 'PLAY' COMMAND

COMMENTS LINE BY LINE

| LINE | COMMENT |
|-------|--|
| 1 | BEGIN A NEW BLOCK WITH LABEL 'STORS' ON MT AND NAME 'STORS SRC' ON THE DT FILE ON UNIT 5. |
| 2-4 | ASSIGN FREQUENCY VALUES TO THE SYMBOLS G0, A0 ETC. |
| 5 | NOT1, NOT2 ARE TIME VALUES OF A WHOLE NOTE RESP. A HALF NOTE. MAX IS USED IN ENV TERMS ON LINES 10,13,15. WF IS WAVEFORM FOR ALL THE 'FG':S |
| 6 | ET1,ET2 ARE CURVEFORMS OF THE ENV TERMS ON LINES 10,15 START OF DEFINITION OF THE 'TON' MACRO. THE TONE STARTS AT GT1 SEC. GT2 LOCAL TIME. MAX1 IS MAXIMUM AMPLITUDE OF THE FIRST FG. T1= 1/10 , T2=8/10 T3=2/10 OF THE WHOLE TONE TIME 'TID'. 602 MS S/4 |
| 7 | ADD THE TONE TIME 'TID' TO GT2. IF GT2 MORE THAN 1000MS (1 SEC.) THEN GT3 WILL BE POSITIVE AND 1 IS ADDED TO GT1 (1 SEC.) AND 1000 MS (1 SEC.) I.E. THE SAME AMOUNT IS SUBTRACTED FROM GT2. |
| 8 | FR1 IS THE FREQUENCY USED IN THE GLIS TERM AT LINE NR 11. |
| 9 | IF THE VARIABLE 'LISTA' IS DEFINED THE VALUES OF MAX1,T1,T2,T3,GT1,FR1 IS WRITTEN OUT. AFTER THAT THE MESSAGE '???' IS WRITTEN OUT AND THE PROGRAM WAITS IN ERROR MODE FOR INPUT FROM UNIT 4 (TT OR TV). A 'DELETE(LISTA)' WILL MAKE THE VARIABLE LISTA UNDEFINED AND THUS MAKE THAT NO MORE OF THE ABOVE PRINTOUTS WILL COME OUT. |
| 10 | THE FIRST FG GOES FROM 50 TO MAX1 DB IN T1 MS CURVEFORM 2, FROM MAX1 TO MAX IN T1 MS, CURVEFORM ET2 AND STAYS AT MAX DB FOR T2 MS. |
| 11 | FIRST FG GOES FROM FR1 TO FR HZ IN T2 MS. |
| 12 | FR1 IS THE FREQUENCY OF THE FIRST OVERTONE. NR1 IS THE NUMBER OF THE SECOND FG. MAX1 IS USED IN THE ENV TERM ON LINE 15. |
| 13 | THE SECOND FG GOES FROM 50 TO MAX DB IN T1 MS, CURVEFORM 0 AND FROM MAX TO 40 DB IN T2 MS. |
| 14 | FR1 IS THE FREQUENCY OF THE THIRD OVERTONE USED IN THE 'TON' MACRO. NR1 IS NUMBER OF THE THIRD FG. MAX2 IS USED IN THE ENV TERM ON LINE 15. |
| 15 | THE THIRD FG GOES FROM 50 TO MAX1 DB IN T1 MS, CURVEFORM ET1, FROM MAX1 TO MAX2 DB IN T1 MS AND FROM MAX2 TO 30 DB IN T2 MS. |
| 16 | START OF THE DEFINITION OF THE 'SILL' MACRO WHICH CONTAINS THE WHOLE CANON "STORSILL OCH SMA SILL". |
| 16-30 | TIME AND FREQUENCY OF THE TONES IN THE CANON ARE GIVEN AT EACH LINE. NOTE THAT THE LOCAL TIME IS ADDED WITH THE TIME VALUE OF THE NOTE EACH TIME THE MACRO 'TON' IS CALLED SO THAT THE TONES WILL COME IN THE RIGHT TIME SEQUENCE. |
| 31 | END OF THE 'SILL' MACRO. |
| 32-39 | FG:S 1-6 ARE CONNECTED TO CHANNEL 1,FG:S 7-12 TO CHANNEL 2, FG:S 13-18 TO CHANNEL 3 AND FG:S 19-24 TO CHANNEL 4. |
| 40 | STARTTIME 1 SEC. AND 0 MS FOR THE FIRST 'SILL' MACRO. |
| 41 | FG:S 1-3 PLAYS THE CANON TWICE. |

42 FG:S 7-9 PLAYS THE CANON TWICE STARTING AT LOCAL TIME 5 SEC. 0 MS.
43 FG:S 13-15 PLAYS THE CANON TWICE STARTING AT LOCAL TIME 9 SEC. 0 MS.
44 FG:S 19-21 PLAYS THE CANON TWICE STARTING AT LOCAL TIME 13 SEC. 0 MS.
45 TRANSFER THE SORTRECORDS PRODUCED BY THE ABOVE 'SILL' MACROS ON THE
'TEMP' DISK TO THE 'MIX' DISK. ("TRANSFER" HERE ALSO INCLUDES SORT)
46 MAKE MT CODE FROM THE MIX DISK. ERASE THE MIX AND TEMP DISKS UNLESS ^{COMMAND.}
ANY OF THE COMMANDS KEEP, KEPP(MIX), KEEP(ALL) COMES AFTER THE END COMMA
THE PROGRAM NOW EXPECTS A NEW BEGIN COMMAND TO START A NEW BLOCK.

.EJECT