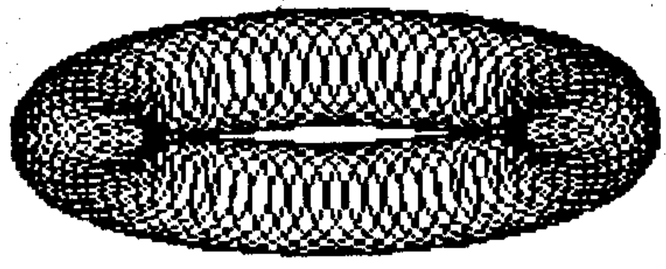
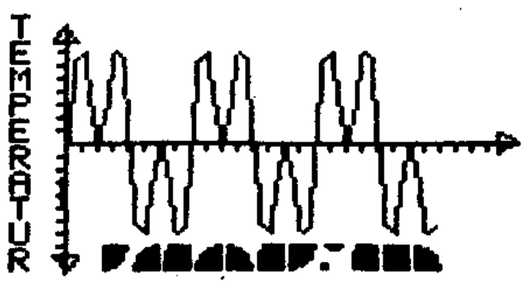
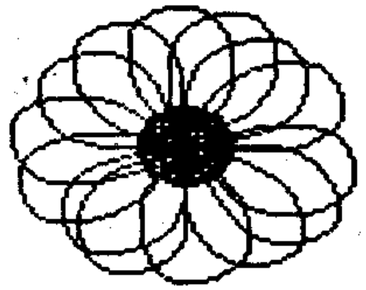
Amerisoft International Presents

EXPANDED GRAPHICS BASIC

CREATE INCREDIBLE GRAPHICS FROM BASIC WITH 30 NEW COMMANDS.







This program requires a TI-99/4A Home Computer, Memory Expansion, at least one Disk Drive, and either an Extended Basic or Editor-Assembler Command Module.

Try Our Other Fine Programs
Including

MINI-EDITOR

"SPEEDGRAPH 99"

TIBB'S (TM)

VOID

HEIST

SPRITEMAKER

Sneaky Snake

GRAPHICS GRABBER

"MASTER PAINTER 99"

"DISK SURGEON 99"

"COPY-CAT (TM)"

3D-WORLD

And Many Others
To Be Released Soon

CONTENTS

1.) INTRODUCTION

AMERISOFT EXPANDED GRAFIC BASIC meets the desire of the TI-99/4A owner for a High Resolution Graphic! With AMERISOFT EXPANDED GRAFIC BASIC, just called AMERISOFT XGBASIC, the powerful graphic capabilities of the computer are displayed.

The addressing of each pixel of the screen becomes possible, all this in 16 fore- and background colors! 40 powerful graphic commands are available and can be called upon by TI-BASIC or TI-EXTENDED BASIC.

The graphic output through a matrix printer is included by software, too. The generating of the graphic works superfast. All the routines are written in the 16 bit machine code TMS 9900 ASSEMBLER. It is creative and fun to watch AMERISOFT XGBASIC at work.

Machine programs cannot be executed by the TI-99/4A keyboard alone. For the application of AMERISOFT XGBASIC one of the following hardware-configurations is essential:

TI-99/4A + MINI MEMORY module
+ Floppy disk system or cassette recorder

TI-99/4A + EDITOR/ASSEMBLER module
+ 32 K RAM expansion
+ Floppy disk system

TI-99/4A + EXTENDED BASIC module
+ 32 K RAM expansion
+ Floppy disk system

The graphic routines are delivered together with graphic demo programs either on floppy disks or cassettes.

2.) PRINCIPLE OF AMERISOFT XGBASIC

2.1) WORKING MODE OF AMERISOFT XGBASIC

The principle of AMERISOFT XGBASIC is very simple. The cursor is a graphic plotter. The screen is the drawing sheet, and when initiating the "GRAFIC MODE" the cursor or drawing pen will appear at position 1, 120 in the drawing window (left bottom corner), ready to start operating along the horizontal axis at the angle of 0 degrees. Simultaneously, it is the 0-point of the system of user-defined co-ordinates.

By means of simple graphic commands the cursor is directed across the screen (see Fig. 1). All co-ordinates are identical to Mathematical ones. Thus lines, circles, squares, ellipses, arcs and many other figures can be drawn or erased. Complex geometrical figures, diagrams, and so on can be generated on the screen at a very high speed. All this with the resolution of 256 * 192 pixels in all the colors available to the TI-99/4A. These graphics can be stored on a floppy disk for later use.

With AMERISOFT EXPANDED GRAFIC BASIC you have purchased a powerful programming-tool for the interesting realization of your programs.

2.2) THE SCOPE OF AMERISOFT EXPANDED GRAFIC BASIC

AMERISOFT XGBASIC is available either on floppy disk or on cassette. On these the following programs are stored:

i) XGBASIC in TMS 9900 ASSEMBLER

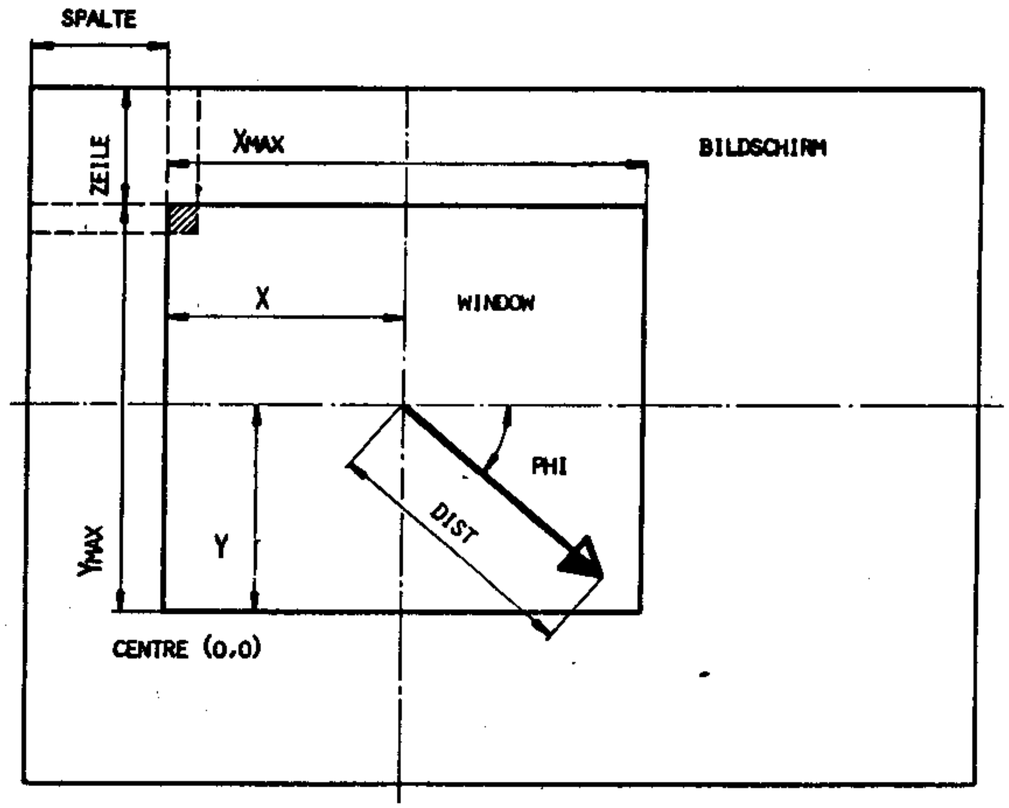
ii) "XGBSCDEMO", demonstration-program in TI-BASIC or TI-XBASIC

iii) AMERISOFT XGBASIC Manual

AMERISOFT XGBASIC is simple and easy to learn.

```
> CALL LINK("WINDOW",ROW,COLUMN)
> CALL LINK("SETTO",X,Y)
> CALL LINK("TURNTO",PHI)
> CALL LINK("MOVE",DIST)
```

Fig. 1.): Principle of AMERISOFT XGBASIC



3.) LOADING OF AMERISOFT XGBASIC

For interfacing AMERISOFT XGBASIC with TI-BASIC or TI-EXTENDED BASIC, three versions of AMERISOFT XGBASIC are offered, the choice of which depends on the hardware-configuration.

3.1) XGBASIC FOR MINI MEMORY MODULE

This is the simplest and most inexpensive graphic version of AMERISOFT XGBASIC.

3.1.1) HARDWARE-CONFIGURATION

- TI-99/4A keyboard
- MINI MEMORY module
- Cassette recorder and/or floppy disc system

3.1.2) LOADING OF XGBASIC FROM CASSETTE

i) Switch off keyboard
ii) Insert MINI MEMORY module
iii) Switch on keyboard
iv) Choose "EASY BUG" from "MASTER TITLE SCREEN"
v) Insert AMERISOFT XGBASIC cassette in CS1
vi) Press key "L" (LOAD STORAGE FROM CS1)
vii) Proceed only according to commands given on the screen. XGBASIC1 starts at position 000 of the cassette. XGBASIC2 starts at position 040 of the cassette.
viii) After loading the program finish with "FCTN QUIT" closing "EASY BUG"
(Should anything go wrong with the hardware repeat steps from iv) inclusively
ix) Choose TI-BASIC from the main menu

3.1.2.1) CHARACTERISTIC FEATURES OF XGBASIC WITH CASSETTES

If no floppy disc system is available, the commands

> CALL FILES(9)
> NEW

cannot be carried out. Since the TI-BASIC program will be destroyed when executing the first AMERISOFT XGBASIC command without the sequence of these commands, the presence of a disc system must be feigned with the commands:

> CALL LOAD (-31888,43,179)
> NEW

The commands must be carried out before the first BASIC-command or the first BASIC program is fed.

These commands

> CALL FILES(9)
> NEW

must be omitted.

3.1.3) LOADING OF XGBASIC FROM FLOPPY DISCS

i) Switch off TI-99/4A keyboard
ii) Insert MINI MEMORY module
iii) Switch on floppy disc system
iv) Switch on keyboard
v) Choose TI-BASIC from the main menu
vi) Insert AMERISOFT disc into disc drive 1
vii) Enter
> OLD DSK1.XGBSCLOAD1
> RUN
or enter
> OLD DSK1.XGBSCLOAD2
> RUN
viii) Enter
> CALL FILES(9)
> NEW
ix) The computer is now ready for graphic programs.

3.1.4) SCOPE OF XBASIC FOR MINI MEMORY MODULE

Because of the limited storage capacity of the MINI MEMORY module XGBASIC contains 2 subsets:

Subset 1 : XGBASIC1 consists of:

GRAFIC, WINDOW, SETBLE, CENTRE, CLSCRN, CLTBLE, TABLE, SETCOL, INVERT, SETTO, RESETO, IFSET, MOVE, REMOVE, MOVETO, REMVTO, TURN, TURNTO, RECT, CLRECT, CIRCLE, CLCIRCLE, ARCUS, CLARCS, ELLIPS, CLLIPS, VALUES, WRITE, BHCOPY, BYEBYE.

Subset 2 : XGBASIC2 consists of:

GRAFIC, WINDOW, SETBLE, CENTRE, CLSCRN, CLTBLE, TABLE, SETCOL, INVERT, SETTO, RESET, IFSET, MOVE, REMOVE, MOVETO, REMVTO, TURN, TURNTO, RECT, CLRECT, CIRCLE, CLCIRCLE, VALUES, WRITE, DSPLAY, ACCEPT, AXIS, HSTDIA, CRCDIA, BHCOPY, BYEBYE.

The statements "DHCOPY", "GSAVE" and "GLOAD" are not included in the XGBASIC-version of MINI MEMORY MODULE.

3.1.5) NOTE

If the XGBASIC is loaded into the 4 K RAM of the MINI MEMORY, it will remain in the memory after switching off the computer. XGBASIC will have to be reloaded, if the command

```
> CALL INIT
```

is executed or the TMS 9900 ASSEMBLER program is overwritten.

3.2) XGBASIC FOR EDITOR/ASSEMBLER MODULE

This version of XGBASIC is sophisticated in every respect and will leave hardly any of your wishes unanswered.

3.2.1) HARDWARE-CONFIGURATION

- TI-99/4A keyboard
- EDITOR-ASSEMBLER module
- System expansion box
- 32 K Memory expansion
- Floppy disc system

3.2.2) LOADING OF XGBASIC

i) Insert EDITOR/ASSEMBLER module
ii) Switch on disc drive
iii) Switch on system expansion box
iv) Switch on keyboard
v) Insert AMERISOFT disc
vi) Choose TI-BASIC from the main menu
vii) Enter

```
> OLD DSK1.XGBSCLOAD
> RUN
```

upon which XGBASIC will be loaded automatically

viii) Enter

```
> CALL FILES(9)
> NEW
```

ix) The TI-BASIC is now ready for the entry of graphic programs.

3.2.3) NOTE

If
```
> CALL FILES(9)
> NEW
```

is not executed before the TI-BASIC program is entered, the BASIC-program in the range of the lower line numbers will be destroyed when executing the program.

Only switching off, waiting a short time and switching on again can initialize the computer.

Caution:

In connection with XGBASIC the maximum number of open files is limited to two.

3.3) XGBASIC FOR XBASIC MODULE

This version of AMERISOFT XGBASIC is the same as XGBASIC for EDITOR/ASSEMBLER module.

Due to the different module structure, different versions of TMS 9900 are necessary.

3.3.1) HARDWARE-CONFIGURATION

- TI-99/4A keyoard
- EXTENDED BASIC module
- System expansion box
- 32 K RAM expansion
- Floppy disc drive

3.3.2) LOADING OF XGBASIC

i) Insert XGBASIC module
ii) Switch on disc drive
iii) Switch on expansion box
iv) Switch on keyboard
v) Insert AMERISOFT disc in disc drive 1
vi) Choose XBASIC from main menu
vii) XGBASIC is loaded automatically
(the name of the load-program is LOAD)
viii) Enter
> CALL FILES(9)
> NEW
ix) The XBASIC is now ready for the entry of graphic programs.

3.3.3) NOTE

If

> CALL FILES(9)
> NEW

are not entered before executing the XBASIC-program, string variables and XBASIC-interpreter-buffer in the VDP-RAM are overwritten and the operating system of the computer will get out of control.

The computer will not react to any entries and can only be brought to normal performance by switching off, waiting a short time and switching it on again.

Caution:

Due to XGBASIC the number of open files in XBASIC is limited to a maximum of two.

3.3.4) ERROR MESSAGES

Error messages of XBASIC when executing XGBASIC commands will be incorrect, since the responsible buffer-table of the TI-99/4A is used for generating graphics.

Sometimes the TI-99/4A can fail after an interrupt with "FCTN CLEAR" or by choosing a subroutine and can be initialized only by switching off and switching on again after a short period.

3.4) COPYRIGHT

Purchased discs are protected against unauthorised copying. As this machine program initiates the TI-99/4A for the AMERISOFT EXPANDED GRAFIC BASIC mode, incorrect program flows may occur if programs other than original AMERISOFT graphic ones are used.

4.) COMMANDS

AMERISOFT EXPANDED GRAFIC BASIC consists of a number of very powerful commands. They simplify the development of complicated graphics tremendously. They are BASIC support commands and are constituted:

> CALL LINK("PROZEDURNAME",PARAMETER *(,PARAMETER,.....)*)

CALL LINK connects the BASIC and ASSEMBLER program, PROZEDURNAME1 actuates a certain TMS 9900 routine, PARAMETERs are optional.

Terms in *()* can be repeated at random. A block up to 15

parameters can be entered at a time.

Depending on the requirements the following parameters are permissable:

- Numerical constants
- Numerical variables
- Numerical array-elements
- Numerical terms
- String constants
- String variable
- String array-elements
- String terms

Every conventional error will be detected and signaled by the TI-BASIC interpreter.

Sometimes there are problems with EXTENDED BASIC, if the graphic table overwrites buffers of the interpreter. It is not always like this, but mostly the given row numbers are not right.

4.1) CALL LINK("GRAFIC",MODUS)

This command signals to the computer the graphic mode and initializes all its registers.

A graphic table with maximal 128 vertical and maximal 120 horizontal lines is defined. This table-section is available for AMERISOFT EXPANDED GRAFIC BASIC for random addressing of every pixel.

The restriction to 128 * 120 = 15.360 pixels is necessary as otherwise there would remain not enough storage capacity in the VDP-RAM of the keyboard for BASIC-programs, string-variables and so on.

In addition the direct addressing of 192 * 256 = 50.176 pixels needs the storage of the 12 K VDP-RAM; character- and color-tables overwrite the buffer-addresses of the BASIC-interpreter.

Since the graphic table cannot be transmitted on the

screen at random, nevertheless there are 256 * 196 = 50.176 pixels to be addressed individually.

CALL LINK(GRAFIC",MODUS) defines the following internal parameters:

PHI=0	Starting angle of the cursor
TABLEWIDTH	16
foreground color	Green
background color	Black

Some more internal parameters are depending on MODUS:

MODUS = 0 : Graphic mode (255 rows for grafic)
15 * 16 table-rows and -columns

X=1, Y=120 : Starting position of the cursor
(0,0)-point

MODUS >< 0 : Text mode (192 rows for graphic)
12 * 16 table-rows and-columns
The commands "DSPLAY" and "ACCEPT" are available for inputut/output-operations.

X=1, Y=88 : Starting position of the cursor
(0,0)-point

CALL LINK("GRAFIC", MODUS) must be the first command before XGBASIC is entered.

As soon as this command is executed the standard characters are lost. The conventional screen 1/0-commands do not lead to the expected results.

CALL LINK("BYEBYE") (see item 4.21) cancels the XGBASIC status and the 1/0 commands work as usual.

Example:

```
100 REM SPIRAL
110 REM ******
```

```
120 CALL LINK("GRAFIC",0)
125 CALL LINK("WINDOW",3,8)
130 CALL LINK("SETTO",64,60)
140 FOR DIST=5 TO 50 STEP 5
150 CALL LINK("MOVE",DIST)
160 CALL LINK("TURN",90)
170 NEXT DIST
180 GOTO 180
```

draws a pale green square-shaped spiral on a black background.

With BREAK ("FCTN CLEAR") the program is interrupted and the standard status of the computer restored.

```
> CONTINUE
```

however does not lead back to the XGBASIC mode, unless the first command following > CON is CALL LINK ("GRAFIC",MODUS)!

4.2) WINDOW

This command transmits sections of the grafic-table or the complete grafic-table to the screen and contains 2 formats.

4.2.1) CALL LINK("WINDOW",COLUMN,ROW)

The graphic window (16*8 columns, 15*8 rows) is set at the screen position column (0-32) and row (0-24). The graphic window can be positioned partly or totally outside the screen (24 rows, 32 columns). Every defined graphic window before

```
> CALL LINK("WINDOW",COLUMN,ROW)
```

will be deleted, if one of both parameters are negative. Only the absolute terms of the parameters are evaluated.

4.2.2) CALL LINK("WINDOW",Z,S,ZA,SA,DZ,DS)

This command transmits sections of the graphic table to the screen. The parameters mean:

Z,S ..... Screen-row (Z) and -column (S) at which the upper left corner-point of the graphic table is projeeted.

ZA,SA ... Upper left corner-point of the graphic table where the projection will start.

DZ ..... Number of characters of the graphic table in direction of rows

DS ..... Number of characters of the graphic table in direction of columns

If Z or S or both are negative, every graphic window which is on the screen will be deleted before the new section is transmitted.

Example:

```
100 REM CIRCLES
110 REM *******
120 CALL LINK("GRAFIC",0)
130 CALL LINK("WINDOW",1,1)
140 FOR R=2 TO 42 STEP 2
150 CALL LINK("CIRCLE",64,60,R)
160 NEXT R
170 CALL LINK("WINDOW",12,18)
175 CALL LINK("WINDOW",1,19)
180 FOR I=1 to 1000
190 NEXT I
200 CALL LINK("SETCOL",16,5)
210 FOR I=1 TO 500
220 NEXT I
230 CALL LINK("INVERT",1,1,128,120)
240 CALL LINK("WINDOW",4,-8)
250 GOTO 250
```

At first a number of concentric circles will appear at the left hand top corner of the screen. These will be copied line 170 and 180) downwards to the left and right (tripled). Line 240 will move these circles back to the middle of the screen, and the remaining circles are deleted. On its way the graphic is changing its color; it is inverted.

Example:

```
100 REM PYRAMIDES
110 REM *********
120 CALL LINK("GRAFIC",0)
130 CALL LINK("WINDOW",1,1,1,1,10,10)
140 CALL LINK ("WINDOW",17,1,1,13,17,13)
142 CALL LINK("WINDOW",13,1)
144 CALL LINK("WINDOW",13,17)
150 CALL LINK("TURNTO",45)
160 CALL LINK("SETTO",64,48)
170 FOR A=1 TO 36 STEP 2
180 CALL LINK("RECT",A,A,-A,A,-A,-A,A-A
190 NEXT A
200 GOTO 200
```

In this example, 4 top views of pyramids are drawn simultaneously. The statements in line 130 and 140 are responsible for this. But the upper left window is only a fragment because the statement 130 transmits only a section of the graphic table.

4.3) SETLE/CLTBLE/TABLE

These are commands for generating the graphic table.

4.3.1) CALL LINK("SETBLE",WIDTH)

This command dimensions the graphic table.

WIDTH ... Number of columns of the graphic table
It can be varied from 1 to 32.

191 characters in the text mode and 255 characters in the graphic mode are available for the graphic table.

The height of the graphic table is depending on the width and is calculated:

```
HEIGHT = INT(255/WIDTH)      for graphic mode
HEIGHT = INT(191/WIDTH)      for text mode
```

In this way higher and wider graphics can be generated.

Caution:

A "WINDOW" statement must follow every "SETBLE" statement. The "WINDOW" statement rearranges the screen, otherwise the graphic generation is not all right.

Simultaneously, "SETBLE" defines the center-point of the system of user defined co-ordinates at pixel-position:

```
CENTRX = 1
CENTRY = HEIGHT * 8 = YMAX
```

4.3.2) CALL LINK("CLTBLE")

This command erases the graphic table and also the graphic.But the table-sections which are transmitted by "WINDOW" statements to the screen remain for the input of new graphics.

4.3.3) CALL LINK("TABLE",Z,S,XMAX,YMAX, BYTES)

This command returns the present parameters of the graphic table (-table) to the following variables:

```
Z ..... Number of rows of the table
S ..... Number of columns of the table
XMAX ..... Maximal pixel-columns of the table
YMAX ..... Maximal pixel-rows of the table
BYTES ..... Number of bytes available for the
            graphic
```

Starting with row 1 and column 12 the character bytes in the graphic table are always arranged in ascending order.

The byte number is calculated:

CHAR# = (ZEILE - 1) * WIDTH + ROW - 1

ROW ..... Row of the graphic table
COLUMN .... Column of the graphic table
WIDTH ..... Absolute width of the graphic table
CHAR# ..... Character-byte number of the table

4.4) SETCOL

This command has two formats and defines fore- and background colors of the graphic.

All the 16 colors known in BASIC can be used either as fore- or as background colors. Several different fore- and background colors can be used simultaneously in one graphic.

4.4.1) CALL LINK("SETCOL",FOREGROUND COLOR,BACKGROUND COLOR)

If only two parameters are present in the parameterlist, the fore- and background colors are altered simultaneously in the entire graphic.

4.4.2) CALL LINK("SETCOL",N,FG,BG*(,N1,FG1,BG1,...)*)

If there are more than two parameters, "SETCOL" defines the foreground color (FG) and background color (BG) for the character sets specified by N.

Thereby the character sets for the rows of the graphic windows are defined as follows:

Eight following bytes construct a character set (0-7, 8-15, ... *usw.*).

COLORS FOR ROWS AND COLUMNS OF THE GRAPHIC AREA (WIDTH = 16):

	COLUMNS	
ROWS	**1-7**	**8-16**
1	1	2
2	3	4
3	5	6
4	7	8
5	9	10
6	11	12
7	13	14
8	15	16
9	17	18
10	19	20
11	21	22
12	23	24
13	25	26
14	27	28
15	29	30

Within the specifications of the above mentioned table, the fore- and background colors can be defined at random.

Thus a multitude of color-combinations is possible. Using a parameterlist, up to 5 color sets can be passed.

Example:

```
100 REM LEAF
110 REM ****
120 RANDOMIZE
130 CALL LINK("GRAFIC",0)
135 CALL LINK("WINDOW",3,8)
140 FOR PHI=0 TO 90 STEP 5
150 CALL LINK("SETTO",64,119)
160 CALL LINK("TURNTO",PHI)
170 CALL LINK("MOVE",1.2*PHI)
180 CALL LINK("TURNTO",180-PHI)
190 CALL LINK("SETTO",64,119)
200 CALL LINK("MOVE",1.2*PHI)
210 NEXT PHI
```

```
220 FOR DELAY=1 TO 500
230 NEXT DELAY
240 FG=2+14*RND
250 BG=2+14*RND
260 CALL LINK("SETTO",FG,BG)
270 FOR DELAY=1 TO 500
280 NEXT DELAY
290 CALL LINK("INVERT",1,1,128,120)
300 GOTO 220
```

Line 190 defines fore- and background color of the graphic, line 220 inverts fore- and background color.

4.5) CALL LINK("INVERT",S,Y,DX,DY)

Calling "INVERT" fore- and background color of the graphic are interchanged. The following parameters are necessary:

X,Y ..... Pixel-position of the upper left corner of the graphic section which is interchanged.
DX ..... Column-pixel-position of the section.
DY ..... Row-pixel-position of the section.

Thereby in the section, pixels which are set are deleted and vice versa.

4.6) CALL LINK("CLSCRN")

This command is similar in its effect to 'CALL CLEAR" in BASIC!

It deletes the graphic, the stored internal cursor parameters remain untouched.

Example:

```
100 REM RANDOM STRAIGHT LINES
110 REM **********************
120 CALL LINK("GRAFIC",0)
125 CALL LINK("WINDOW",-3,8)
130 VG =Z+13*RND
140 IF VG<=3 THEN 130
150 CALL LINK("SETCOL",VG,2)
160 FOR I=1 TO 20
170 X=124*RND
180 Y=120*RND
190 CALL LINK("MOVETO",X,Y)
200 NEXT I
210 FOR J=1 TO 250
210 NEXT J
220 CALL LINK("CLSCRN")
230 FOR I=1 TO 500
240 NEXT I
250 CALL LINK("WINDOW",1,1)
260 GOTO 125
```

This program example draws 20 lines in succession choosing direction at random starting from position (1,1) (line 180), deletes the graphic (line 140) and starts the graphic afresh choosing the colors at random.

After a short time, the statement "WINDOW" (line 250) shows that only the screen with line 220 has been erased, but the graphic in the table has remained untouched.

"WINDOW" with a negative parameter (line 125) effects "CLSCRN", before the section of the graphic table is brought to the screen!

4.7) CALL LINK("CENTRE",X,Y)

This statement defines the system of user defined co-ordinates:

X ..... X-co-ordinate of the 0-point in the graphic table
Y ..... Y-co-ordinate of the 0-point in the graphic table

After a "GRAFIC" statement the center point is exact at position (1,120) (left bottom corner of the table). With "CENTRE" this 0-point can be moved optionally, even outside of the table!

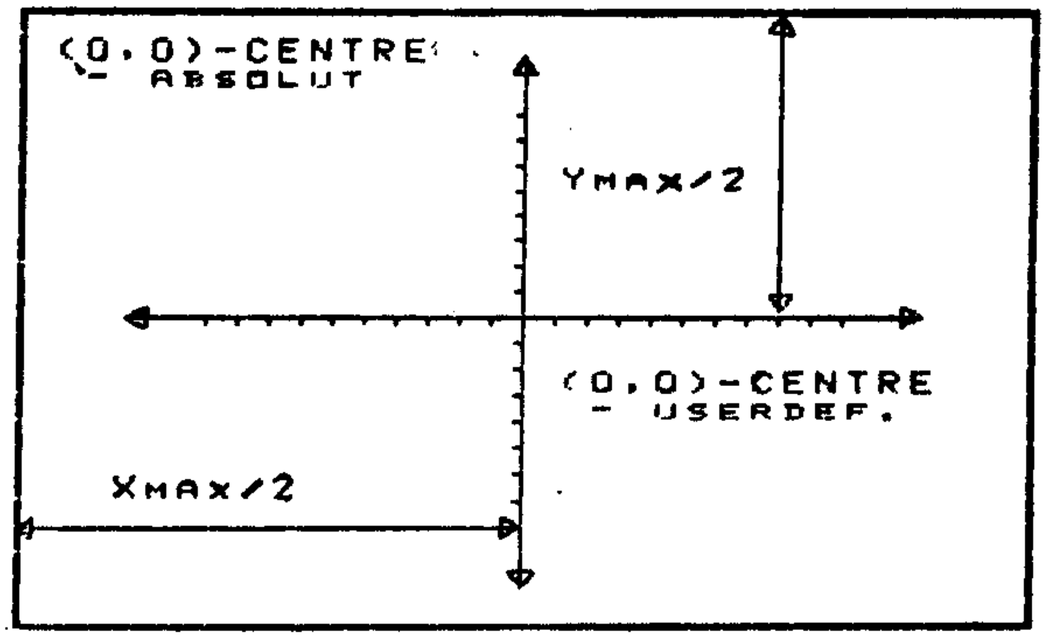


Fig. 2.): System of User-Defined Co-ordinates

Example:

```
100 REM SYSTEM OF CO-ORDINATES
110 REM **********************
120 CALL LINK("GRAFIC",1)
130 CALL LINK("WINDOW",7,8)
132 CALL LINK("SETTO",1,1)
134 CALL LINK("RECT",127,-87)
140 CALL LINK("CENTRE",64,44)
150 CALL LINK("AXIS",0,60,60,4,0,40,40,2)
160 CALL LINK("WRITE",8,10,"(0,0)")
170 CALL LINK("DSPLAY",1,5,26,">CALL LINK(""CEN
TRE"",64,44)")
180 CALL LINK("DSPLAY",5,5,22,"(-64,-44) (64,-44)")
190 CALL LINK("DSPLAY",20,5,22,"(-64,-44) (64,-44)")
200 CALL LINK("BHCOPY",0,"RS232.BA=9600.DA=8.
CR",CHR$(27)&"A"&CHR$(8))
210 STOP
```

This program example produces a drawing for Fig. 4.7.1) for an EPSON-FX-80 or EPSON-RX-80 printer.

4.8) SETTO/RESET/IFSET

4.8.1) CALL LINK("SETTO",X,Y*(,X1,Y1,...)*)

"SETTO" is setting pixels at the coordinates row Y, column X. Columns 1-128 and rows 1-120 are on the screen.

With reference to the range of the values there are no restrictions, the co-ordinates may be positive or negative, its value high or low at random and may also be floating decimal. They are internally rounded to the nearest integer number.

Figures greater than 32.768 and less than -32.767 are displayed incorrectly. There will be no error message when exceeding this range! In using a parameterlist up to 7 pixels can be defined simultaneously. The internal angle PHI of the cursor remains unchanged using "SETTO".

4.8.2) CALL LINK("RESET",X,Y.*(,X1,Y1,...)*)

"RESET" deletes the pixels with the co-ordinates X,Y. All the statements made under "SETTO" are valid for "RESET".

4.8.3) CALL LINK("IFSET",X,Y,VAR*(,X1,Y1,VAR1, ...)*)

This statement checks whether a pixel with the coordinates X,Y is set and the following values in the variables VAR are stated:

Pixel (X,Y)	set	VAR=-1
Pixel (X,Y)	deleted	VAR= 0
Pixel (X,Y)	outside the graphic window	VAR=+1

With parameterlist up to 5 pixels can be checked simultaneously. Otherwise the conditions set up for "SETTO" and "RESET" apply.

4.8.4 Example

```
100 REM SINE
```

```
110 REM ****
120 CALL LINK("GRAFIC",0)
125 CALL LINK("WINDOW",3,8)
127 CALL LINK("CENTRE",4,60
130 PI180=4*ATN(1)/180
140 CALL LINK("AXIS",0,0,118,5,0,50,50,5)
150 REM
160 FOR PHIM0 TO 360 STEP 2
170 X=PHI/3+20
180 Y=20*SIN(PHI*PI180)
190 CALL LINK("SETTO",X,Y,X,Y,*1.5,X,Y,*2)
200 NEXT PHI
210 FOR PHIM0 TO 360 STEP 2
220 X=PHI/3+20
230 Y=20*SIN(PHI*PI180)
240 CALL LINK("RESET",X,Y,*1.5,X,Y*2)
250 NEXT PHI
260 X=INT(128*RND)+1
270 Y=INT(120*RND)+1
280 CALL LINK("IFSSET",X,Y,A)
290 IF A=0 THEN 260
300 CALL LINK("BYEBYE")
310 CALL CLEAR
320 PRINT "PUNKT X=";X;"Y=";Y;":IST GESETZT"
330 STOP
```

This program example draws 3 curves of sines (160-200), deletes it again (210-250) and stays in a holding loop until it has found a set pixel (280-290).

4.9) MOVE/REMOVE

4.9.1) CALL LINK("MOVE",DIST)

This statement draws a line of the length DIST, starting from the present internal angle PHI. The position DIST horizontally and vertically corresponds exactly with the number of pixels; the number of pixels depends on the set angle. After performing DIST the cursor stops at the end coordinates (last stored pixel). PHI remains unchanged.

Positive values of DIST work in the present direction of the cursor, negative values work 180 degrees opposite. DIST can assume any value, although the range limits apply given under item 4.8.1).

4.9.2) CALL LINK("REMOVE",DIST)

"REMOVE" has the same effect as "MOVE", but here the pixels from position X,Y up to the DIST distant new position of the cursor are deleted.

Example

```
100 REM STAR
110 REM ****
120 CALL LINK("GRAFIC",0)
125 CALL LINK("WINDOW",3,8)
130 CALL LINK("WRITE",15,3,
"AMERISOFTGRAFIC")
140 FG=3+13*RND
150 CALL LINK("SETCOL",FG,2)
160 CALL LINK("SETTO",2,60)
170 CALL LINK("TURNTO",36)
180 FOR I=1 TO 10
190 CALL LINK("TURN",108)
200 CALL LINK("MOVE",60)
210 NEXT I
220 CALL LINK("SETTO",2,60
230 CALL LINK("TURNTO",36)
240 FOR I=1 TO 10
250 CALL LINK("TURN",108)
260 CALL LINK("REMOVE",60)
270 NEXT I
280 GOTO 140
```

Through skilled application of a few commands, a star is charmed onto the screen, then deleted and then the game starts anew in different colors.

4.10) MOVETO/REMVTO

4.10.1) CALL LINK(MOVETO",X,Y*(,X1,Y1,...)*)

"MOVETO" draws a line from the present internal position of the cursor to the next by the parameter pair X and Y defined position.

The list of parameters can hold a maximum of 7 positions. If there are more than 2 parameters given, the line will always be drawn from the previous position to the next.

After executing "MOVETO" the cursor will remain at the last position given on the parameterlist. The internal angle and the colors will remain unchanged with "MOVETO".

Lines can also be drawn out of the graphic window.

4.10.2) CALL LINK("REMVTO",X,Y,*(,X1,Y1,...)*)

"REMVTO" works like "MOVETO" with the difference that here the lines are deleted.

All the conditions for "MOVETO" also apply to "REMVTO".

Example

```
100 REM THREAD GRAPHIC
110 REM ***************
120 CALL LINK("GRAFIC",0)
125 CALL LINK("WINDOW",3,8)
130 D=2.6
140 Z=125
150 S1=128
160 FOR S=1 TO 126 STEP 5
170 Z=Z-5
180 S1=S1-D
190 CALL LINK("SETTO",S,120)
200 CALL LINK("MOVETO", S1,Z)
210 NEXT S
220 Z=125
230 S1=1
240 FOR S=126 TO 1 STEP -5
250 Z=Z-5
```



```
260 S1=S1+D
270 CALL LINK("SETTO",S,120)
280 CALL LINK("MOVETO",S1,Z)
290 NEXT S
300 GOTO 300
```

4.11) TURN/TURNTO

4.11.1) CALL LINK("TURN",PHI)

This command adds to the present internal angle of the cursor the angle PHI in decimal degrees. The limits of the angle are +/- 2047 degrees. Internally, the angle is modulated from 0-360 degrees.

The trigonometrical functions are generated by the computer via interpolation tables. Since especially with MINI MEMORY module the storage capacity is very limited, the angles are interpolated in the range of 5 degrees. This may lead to inexact results when using intermediate values.

4.11.2) CALL LINK("TURNTO",PHI)

This command imperatively sets the internal angle of the cursor to PHI (degrees). All limitations made for "TURN" also apply here.

Example

```
100 REM OCTAGONS
110 REM ********
120 CALL LINK("GRAFIC",0
130 CALL LINK("WINDOW",4,5)
140 CALL LINK("WINDOW",6,19)
150 DIST=2
160 FOR S=28 TO 108 STEP 4
170 CALL LINK("SETTO",S,42)
180 CALL LINK("TURNTO",90)
190 DIST=DIST+2
200 FOR I=1 TO 8
210 CALL LINK("TURN",45)
220 CALL LINK("MOVE",DIST)
```

```
230 NEXT I
240 NEXT S
250 GOTO 250
```

4.12) RECT/CLRECT

4.12.1) CALL LINK("RECT",A,B*(,A1,B1,...)*)

Starting from its present position and the internal angle of the cursor "RECT" draws rectangles with the sequence

A -> B -> A -> B

The rectangle turns clockwise, if B is positive. The Example shows the influence of the operational sign of the side length with reference to its ultimate position.

The internal angle and the position of the cursor are not influenced by "RECT". The side length of the rectangle can take any value. Up to 7 rectangles can be passed with one parameterlist. But they all begin at the same starting position and also finish there.

4.12.2) CALL LINK("CLRECT",A,B*(,A1,B1,...)*)

Works completely identical to "RECT" with the difference that "CLRECT" deletes the rectangles. All conditions of "RECT" apply to "CLRECT".

Example

```
100 REM RECTANGLES
110 REM **********
120 CALL LINK("GRAFIC",0)
130 CALL LINK("SETTO",64,60)
140 A=40
150 B=20
160 CALL LINK("RECT",A,B,A,-B,-A,B,-A,-B)
170 FOR I=1 TO 250
180 NEXT I
190 CALL LINK("CLRECT",A,B,A,-B,-A,B,-A,-B)
200 CALL LINK("TURN",45)
210 GOTO 160
```


Line 160 draws 4 rectangles with only one command, line 190 deletes them. Line 200 turns the internal angle on by 45 degrees.

4.13) CIRCLE/CLCRCL

4.13.1) CALL LINK("CIRCLE",X,Y,R*(,X1,Y1,R1,...)*)

"CIRCLE" draws a circle with the central-point X, Y and the radius R. With one parameterlist, up to 5 different circles can be drawn.

The parameters can assume any value. For the radius the absolute value is worked out automatically. If the value for the radius = 0, "CIRCLE" sets a point (pixel). After executing "CIRCLE", the cursor takes the central point of the last drawn circle. The internal angle PHI remains unchanged. Due to internal rounding errors, the circular arcs may appear not quite smooth.

4.13.2) CALL LINK("CLCRCL",X,Y,R*(,X1,Y1,R1,...)*)

"CLCRL" works identically to "CIRCLE", but here the circles are deleted. All the conditions for "CIRCLE" also apply to "CLCRCL".

Example

```
100 REM CIRCLES
110 REM *******
120 PI=4*ATN(1)
130 CALL LINK("GRAFIC",0)
135 CAll LINK("WINDOW",3,8)
137 CALL LINK("CENTRE",64,60)
140 CALL LINK("CIRCLE",0,0,30)
150 FOR PHI=0 TO 2*PI STEP PI/16
160 CALL LINK("CIRCLE",30*COS(PHI),30*SIN(PHI),
30)
170 NEXT PHI
180 GOTO 180
```

4.14) ARCUS/CLARCS

4.14.1) CALL LINK("ARCUS",X,Y,R,PHI,DPHI*(,X1, Y1,R1,PHI1,DPHI1)*)

"ARCUS" draws circular arcs with the following parameters:

X,Y ..... Centre-point of the arc
R ..... Radius of the ard
PHI ..... Starting angle of the arc (absolute)
DPHI ... Arc-angle of the arc

Simultaneously three arcs can be generated with one command. Due to internal rounding errors and generating the trigonometrical functions via interpolation tables, the results are not always satisfying.

The co-ordinates of the cursor describe the last drawn arc-pixel after executing "ARCUS".

4.14.2) CALL LINK("CLARCS",X,Y,R,PHI,DPHI*(, X1,Y1,R1,PHI1,DPHI1,...)*)

"CLARCS"works identical to "ARCUS", the difference being that "CLARCS" deletes all arc-pixels.

4.15) ELLIPS/CLLIPS

4.15.1) CALL LINK("ELLIPS",X,Y,A,B*(,X1,Y1,A1, B1,...)*)

"ELLIPS" draws ellipses with the axis centre point X,Y of the big semi-axis A and small semi-axis B and the inclination PHI of the big semi-axis.

A maximum of three different ellipses can be drawn with one parameterlist. The parameters can assume any values except 0. The absolute value is automatically used for the big and small semi-axis.

After the excution of "ELLIPS" the cursor assumes the coordinates of the semi-axis points of intersection. The internal angle PHI remains unchanged.

Through internal rounding errors the elliptical arcs may sometimes not appear quite smooth. This occurs particularly when the main axes are inclined to the horizontal or vertical, because the coordinate transformations are carried out by interpolated trigometrical functions.

4.15.2) CALL LINK("CLLIPS",X,Y,A,B*(,X1,Y1,A1, B1*,...)*)

"CLLIPS" works like "ELLIPS", the difference being that here the ellipses are deleted. All the conditions listed for "ELLIPS" are likewise valid.

Example

```
100 REM CONE
110 REM ****
120 CALL LINK("GRAFIC",0)
125 CALL LINK("WINDOW",3,8)
130 CALL LINK("SETCOL",15,2)
140 M=1
150 A=42
160 B=22
170 FOR Y=81 TO 1 STEP -8
180 IF M=0 THEN 220
190 M$="ELLIPS"
200 REM
210 GOTO 230
220 M$="CLLIPS"
230 CALL LINK(M$,64,Y,A,B)
240 A=A-4
250 B=B-2
260 NEXT Y
270 IF M=1 THEN 140
280 M=0
290 GOTO 150
```

If M=1, a cone is always drawn due to the control commands 270-290. N, a string variable, can also be passed as "PROZEDURNAME". The program is discontinued by entering "FCTN CLEAR".

4.16) CALL LINK("VALUES",X,Y,PHI,FG,BG,BB)

"VALUES"returns the present internal parameters to the variable list.

X ..... Cursor column
Y ..... Cursor row
PHI .... Cursor angle
FG .... Foreground color
BG .... Background color

As the angle is modulated internally, it is always between 0-360 degrees, independent of the previous input.

Example

```
100 REM EXAMPLE
110 REM *******
120 CALL LINK("GRAFIC",0)
125 CALL LINK("WINDOW",3,8)
130 For I=1 to 11
140 CALL LINK("SETTO",64,60)
150 CALL LINK("MOVE",30)
160 CALL LINK("VALUES",X,Y,PHI,FG,BG)
170 CALL LINK("CIRCLE",X,Y,20)
180 CALL LINK("TURN",30)
190 NEXT I
200 GOTO 200
```

From the center point of the graphic window, line 150 draws a line with the length 30. Line 160 determines the final cursor position, line 170 takes this position as center point for a circle with radius 20.

4.17) AXIS/HSTDIA/CRCDIA

AMERISOFT offers three utility programs with EXPANDED

GRAFIC BASIC. These programs are called "AXIS", "HSTDIA" and "CRCDIA".

4.17.1) CALL LINK("AXIS",X,LENXL,LENXR, DELTAX,Y,LENYU,LENYD,DELTAY)

"AXIS" draws a system of coordinates with the folowing parameters:

X,Y ..... Center-point of coordinates
LENXL ..... Left hand side X-semi-axis (length)
LENXR ..... Right hand side X-semi-axis (length)
DELTAX .... Pitch of the X-grid
LENYU ..... Top Y-semi-axis (length)
LENYD ..... Bottom Y-semi-axis (length)
DELTAY .... Pitch of the Y-grid

All values are taken as absolute values. If one of the semi-axis has the value 0, then this semi-axis is not drawn. If the value for the grid equals 0 or more than that of the corresponding semi-axis, no grid will be drawn.

After executing "AXIS" the cursor will assume the position at the center point of the coordinate system. The internal angle PHI is altered. The system of coordinates may not be completely on the screen.

Example

```
100 REM ZYKLOM
110 REM ******
120 CALL LINK("GRAFIC",0)
125 CALL LINK("WINDOW",3,8)
130 PI2=8*ATN(1)
140 CALL LINK("AXIS",8,0,116,4,60,60,59,4)
150 X=7
160 CALL LINK("SETTO",X,40)
170 FOR PHI=0 TO 3*PI2 STEP P12/16
180 X=X+2
190 Y=20*(SIN(2*PHI)+2*COS(PHI))+60
200 CALL LINK("MOVETO",X,Y)
210 NEXT PHI
220 GOTO 220
```

4.17.2) CALL LINK("HSTDIA",X,Y,WIDTH,HEIGTH, DEPTH)

"HSTDIA" draws a block diagram with the following parameters:

X,Y ..... Coordinates of the left bottom corner of the block
WIDTH ..... Width of block diagram
HEIGTH ..... Heighth of block diagram
DEPTH ..... Depth of block diagram

Only the absolute values are taken.

Example

```
100 REM HISTOGRAMS
110 REM **********
120 CALL LINK("GRAFIC",0)
125 CALL LINK("WINDOW",3,8)
130 FOR N=2 TO 20
140 CALL LINK("SETCOL",N,14,2)
150 NEXT N
160 CALL LINK("AXIS",8,10,120,8,20,90,0,4)
170 CALL LINK("HSTDIA",16,22,12,80,6)
180 CALL LINK("HSTDIA",40,22,12,45,6)
190 CALL LINK("HSTDIA",70,22,12,67,6)
200 CALL LINK("HSTDIA",94,22,16,12,6)
210 CALL LINK("WRITE",15,3,"HISTOGRAMS")
220 GOTO 220
```

4.17.3) CALL LINK("CRCDIA",X,Y,RADIUS,PHI, DPHI*(X1,Y1,RADIUS1,PHI1,DPHI1)*)

"CRCDIA" draws a circular diagram with the following parameters:

X,Y ..... Coordinates of circular segment center point
RADIUS .... Radius of circular segment
PHI ..... Starting angle of circular segment
DPHI ..... Finishing angle of circular segment

All values are taken in the absolute.

Example

```
100 REM CIRCULAR DIAGRAMS
110 REM ******************
120 CALL LINK("GRAFIC",0)
125 CALL LINK("WINDOW",3,8)
130 FOR N=1 TO 13 STEP 2
140 CALL LINK("SETCOL",N,16,2,N+1,14,2,N+14,9,2,
N+15,13,2)
150 NEXT N
160 CALL LINK("CRCDIA",64,60,38,180,276)
170 CALL LINK("CRCDIA",68,60,38,276,450)
180 CALL LINK("CRCDIA",68,64,38,90,180)
190 CALL LINK("WRITE",14,2,"CIRC-DIAGRAMS")
200 GOTO 200
```

4.18) WRITE/DSPLAY/ACCEPT

These commands enable the mixing of graphic and text and extend the I/O operations. They can be used in the normal BASIC mode of the computer, too, if the computer is not in the GRAFIC mode.

4.18.1) CALL LINK("WRITE",Z,S,STRING*(,Z1,S1, STRING1,....)*)

"WRITE" enables the mixing of graphic and text.

At position row (Z) and column (S) of the graphic window "WRITE" displays a string (STRING).

Limits: Z .....1 to 15
S .....1 to 16

If the string is too long for the line in question, the rest will be cut.

A maximum of 5 strings can be entered with one parameterlist. The ASCII-codes (capital letters, small letters, digits and special characters) apply.

4.18.2) CALL LINK("DISPLAY".Z.S.SIZE.VARS)

This command corresponds to the well known EXTENDED BASIC command "DISPLAY AT" and uses the following parameters:

Z	Row of screen	(1-24)
S	Column of screen	(1-32)
SIZE	Length of the string	(max. 32)
VAR$	String variable	(max. 32 characters)

With this statement SIZE characters of the string VAR are transmitted to the screen position (Z,S). Thereby graphic values lying under the string are erased (difference to "WRITE"!). If SIZE is negative, SIZE positions are not deleted before the output of the string.

4.18.3) CALL LINK("ACCEPT".Z.S.SIZE.VAR$)

This command corresponds to the well known EXTENDED BASIC command "ACCEPT AT" and uses the following parameters:

Z	Row of screen	(1-24)
S	Column of screen	(1-32)
SIZE	Length of the string	(max. 32)
VAR$	String variable	(max. 32 characters)

"ACCEPT" accepts SIZE characters of a string at the position (Z,S). If SIZE is positive, SIZE positions are deleted previously.

During the input the following keys are active:

UALPHA	ASCII-codes (32-96)
<-	Cursor left
->	Cursor right
ERASE	Deletes the input
ENTER	Accepts the string
REPEAT	Repeat function

CAUTION

"FCTN QUIT" and "FCTN CLEAR" are ineffective while executing this command. While accepting all key codes can be addressed, but for text purposes only UALPHAs (capital letters) are useful.

Example:

```
100 REM STAR1
110 REM *****
CALL LINK("GRAFIC".1)
```



```
125 CALL LINK("WINDOW",3,8)
130 CALL LINK("SETCOL",12,2)
140 CALL LINK("SETTO",5,60)
150 CALL LINK("MOVE",120)
160 CALL LINK("SETTO",64,1)
170 CALL LINK("TURN",90)
180 CALL LINK("MOVE",120)
190 S1=5
200 Z1=60
210 Z2=36
220 S2=64
230 S3=124
240 Z3=60
250 Z4=84
260 S4=64
270 FOR I=1 TO 10
280 CALL LINK("SETTO",S1,Z1)
290 CALL LINK(MOVETO",S2,Z2,S3,Z3,S4,Z4,S1,
Z1)
300 S1=S1+4
310 Z2=Z2-4
320 S3=S3-4
330 Z4=Z4+4
340 NEXT I
350 CALL LINK("WRITE",15,1,"AMERISOFT
GRAFIC")
360 CALL LINK(DSPLAY",17,3,27,"BY ENTERING
OF   STOP")
370 CALL LINK(DSPLAY",18,3,"THE GRAFIC IS
  STOPPED!)
380 CALL LINK("ACCEPT",22,3,4,M$)
390 IF M$<>STOP THE 120
400 STOP
```

4.19) GSAVE/GLOAD

4.19.1) CALL LINK("GSAVE","FILENAME")

Graphics can only be stored on floppy discs. Any valid file name may be used. Storage is formatted in the "memory-image". The colors of the graphic and the position of the graphic window are stored.

4.19.2) CALL LINK("GLOAD","FILENAME")

This statement loads a graphic called "FILENAME" from the disc into the VDP-RAM.

Calling upon "GLOAD" without previous "GRAFIC" will lead to an error message and discontinuation of the program.
Example

```
100 REM STORE GRAPHIC
110 REM *************
120 CALL LINK(GRAFIC",0)
125 CALL LINK("WINDOW",3,8)
130 FOR R=2 TO 40
140 CALL LINK("CIRCLE",64,60,R)
150 NEXT R
160 CALL LINK("GSAVE","DSK2.SAVETEST")
170 STOP

  > NEW

100 REM LOAD GRAPHIC
100 REM ************
120 CALL LINK("GRAFIC",0)
130 CALL LINK("GLOAD","DSK2.SAVETEST")
140 GOTO 140
```

The name of the file may not be longer than 9 characters.

The previous example results in generating the following files in DSK2.:

"SAVETEST"	Contains graphic-area and parameters
"SAVETEST1"	Contains a screen dump
"SAVETEST2"	Contains colors of the graphic

By saving and loading the graphic, only the first file name, in our case "SAVETEST", must be stated.

The input of the second or third file name leads to wrong functions.

The statement "WINDOW" can be cancelled by loading the graphic because every window on the screen is over-written by the loaded graphic. "GSAVE/GLOAD" does not apply to MINI MEMORY module!

4.20) HARDCOPY

In TMS 9900 XGBASIC, 2 hardcopy routines for EPSON formatted printers are contained.

In this way the creation of graphic through the matrix-printer becomes very easy and superfast, for one complete screen-dump only a few seconds are necessary. These hardcopy routines work in the normal BASIC mode of the computer, too.

4.20.1) CALL LINK("DHCOPY",DESC,POP$,PADJ$)

This statement is available only for the EPSON FX-80 printer and produces a screen-dump in the "DOWN LOAD CHARACTERSET MODE". (See the manual of the EPSON FX-80 printer).

DESC	Proportional data for characters (Upper-, under-scoring, proportional-print -> see manual of the FX-80!)
POPT$	printer-options — usually: "RS232.BA+9600.DA+8.CR" The printer must be set with the DIPS in "DOWN LOAD CHARACTERSET MODE".
PADJ	Printer-adjustment f.e.: CHR$(27)& "p" &CHR$(1)&CHR$(27)&"A"&CHR$(8) for proportional-print and 8 dots line feed DESC = 135

In this mode no 1:1 graphic representations can be generated. This command is not available for MINI MEMORY!

4.20.2)CALL LINK("BHCOPY".MOD.POPT$.PADJ$)

"BHCOPY" produces screen dumps through the FX-80 or RX-80 printer in the so-called "BIT IMAGE MODE".

The following parameters correspond to:

MOD ..... 0 to 6 graphics mode of the printer EPSON FX-80, RX-80 according to the EPSON manual.

for MOD = 0 the graphic generation through the printer is 1:1.

MOD 7 —> ESCK	FX-, MX-80, STAR and TI-99 printer
MOD 8 —> ESCL	FX-, RX-, MX-80 and STAR and TI-99 printer
MOD 9 —> ESCY	FX-, RX-80 and STAR printer
MOD 10 —> ESCZ	FX-, RX-80 and STAR printer
MOD 11 —>	SEIKOSHA GP 100 M2 printer
MOD 12 —>	SEIKOSHA GP 550 printer

POPT$ ..... printer-options
usually: "RS232.BA=9600.DA=8.CR"
or "PIO.CR"
The DIPS of the printer must be set correspondingly.

PADJ ..... Printer-adjustment
for example: CHR$(27)&"A"&CHR$(8) for 8 DOTS line feed EPSON printers. Or:
CHR$(27)&"9"&CHR$(13) for SEIKOSHA 550

Example

```
100 REM EPSONCOPY
110 REM *********
120 CALL LINK("GRAFIC",0)
130 CALL LINK("WINDOW",3,8)
140 CALL LINK("CENTRE",64,60)
150 FOR PHI=0 TO 355 STEP 5
160 CALL LINK("SETTO",0,0)
170 CALL LINK("TURNTO",PHI)
180 CALL LINK("MOVE",50)
190 NEXT PHI
200 CALL LINK("WRITE",15,6,"BHCOPY")
210 POPT$="RS232.BA=9600.DA=8.CR"
220 CALL LINK("BHCOPY",0,POPT$,CHR$(27)&
"A"&CHR$(8))
230 REFLF$=CHR$(27)&"j"&CHR$(255)
235 REFLF$=REFLF7$&REFLF$&CHR$(27)&
"j"&CHR$(80)&CHR$(10)
240 CALL LINK("WRITE",15,6,"D")
250 CALL LINK("DHCOPY",135,POPT$,REFLF$&
CHR$(27)&"p"&CHR$(1)&CHR$(27)&"1"&/R$(30)
260 STOP
```

Sometimes while executing "DHCOPY" or "BHCOPY", errors can happen:

UNKNOWN ERROR IN. . .

or

I/O ERROR IN . . .

In this case all complex string terms must be removed from the parameterlist and replaced by simple string terms. See the example "HCOPYDEMO".

```
1680 CALL LINK("BHCOPY",0,POPT$,PADJ$)
1690 REM
1700 REM BLUMEN
1710 REM ******
1720 CALL LINK("GRAFIC",0)
1730 CALL LINK("WINDOW",1,4)
1740 CALL LINK("CENTRE",54,48)
1750 CALL LINK("SETTO",0,0)
1760 FOR PHI=0 TO 22.5 STEP 22.5
1770 CALL LINK("TURNTO",PHI)
1780 FOR X=1 TO 7
1790 FOR W=0 TO 15
1800 CALL LINK("MOVE",4)
1810 CALL LINK("TURN",W)
1820 NEXT W
1830 FOR W=16 TO 4 STEP -1
1840 CALL LINK("MOVE",4)
1850 CALL LINK("TURN",W)
1860 NEXT W
1870 CALL LINK("MOVETO",0,0)
1880 CALL LINK("TURN",6)
1890 NEXT X
1900 NEXT PHI
1910 FOR R=1 TO 12
1920 CALL LINK("CIRCLE",0,0,R)
1930 NEXT R
1940 PADJ$=LM$(50)*RLF$(240)&LF8$
```

HCOPYDEMO (PART 6)

```
1950 CALL LINK("BHCOPY",0,POPT$,PADJ$) 373,
377,388
1970 OPEN #1:POPT$,OUTPUT
1980 PRINT #1:RLF$(2);LF$;LF$;LF$;LF$;LF$;
EMPON$;LM$(12);"E X P A N D E D";LF$;L F$
1990 PRINT #1:LF$;LM$(6);" G R A F I C B A S I C";LF$
2000 PRING #1:CHR$(27);"@"
2010 CLOSE #1
2020 CALL LINK("BYEBYE")
2030 END
```


CALL LINK ("BYEBYE")

"BYEBYE" cancels the GRAFIC mode. It loads the standa character sets and establishes the BASIC or the XBAS mode.

The computer works as usual. SOUND and SPRITES can l used again. Before executing a new AMERISOF EXPANDED GRAFIC BASIC statement a CALL LIN ("GRAFIC",MODUS) command must be entered onc more, otherwise the program is discontinued with an err message.

5.) QUICK REFERENCE

CALL LINK("GRAFIC",MODUS)

signals the graphic mode, initializes all the compute registers and defines a graphic table (max. 128 vertical an 120 horizontal lines) depending on the MODUS (graphic c text mode.)

CALL LINK("WINDOW",ROW,COLUMN)

sets the entire graphic table (16*8columns, 15*8 rows o 11*8 rows in the text mode) at position column (1-32) anc row (1-24).

CALL LINK("WINDOW",Z,S,ZA,SA,DZ,DS)

transmits sections of the graphic table to the screen; Z means the row, S the column, ZA and SA mean the upper left corner-point of the graphic table, DZ the number of graphic table characters in row-direction and DS the number of graphic table characters in column-direction.

CALL LINK("SETBLE",WIDTH)

dimensions the graphic table.

CALL LINK("CLTBLE")

clears the graphic table and thus the graphic

CALL LINK("TABLE",Z,S,XMAX,YMAX,BYTES)

returns the present parameters of the graphic table to the given variables, whereby Z corresponds to the number of rows, S to the number of columns, XMAX to the maximal pixel-columns, YMAX to the maximal pixel-rows of the table and BYTES to the number of available bytes.

CALL LINK("SETCOL", FOREGROUNDCOLOR, BACKGROUNDCOLOR)

changes the fore- and background-color simultaneously for the entire grafic.

CALL LINK("SETCOL",N,FG,BG*(N1,FG1,BG1,...)*)

defines the foreground-color (FG) and the background-color (BG) for the characterset specified by N.

CALL LINK("INVERT",X,Y,DX,DY)

inverts sections of the graphic; with X, Y the pixel-position of the upper left corner, with DX the pixel-column position and with DY the pixel-row position of the inverted graphic-section are stated.

CALL LINK("CLSCRN")

deletes the screen the graphic in the tble and all other internal parameters remain.

CALL LINK("CENTRE",X,Y)

defines the system of user-defined co-ordinates with the (0,0)-point at the position (X,Y) of the graphic table.

CALL LINK("CENTRE",X,Y)

defines the system of user-defined co-ordinates with the (0,0)-point at the position (X,Y) of the graphic table.

CALL LINK("SETTO",X,Y*(,X1,Y1,...)*)

sets pixel with the co-ordinates row X and column Y.

CALL LINK("RESET",X,Y,*(,X1,Y1,...)*)

erases pixels with the co-ordinates X, Y.

CALL LINK("IFSET",X,Y,VAR*(,X1,Y1,VAR1,...)*)

examines if a pixel with the co-ordinates X, Y is set or not and returns the result to the variable VAR.

CALL LINK("MOVE",DIST)

draws a line of the length DIST, starting from the present position X, Y of the cursor with the present internal angle PHI.

CALL LINK("REMOVE",DIST)

deletes all pixels from position X,Y up to the DIST distant new position of the cursor.

CALL LINK("MOVETO",X,Y*(,X1,Y1,...)*)

draws a line from the present internal position of the cursor to the next by the parameter pair X,Y defined position.

CALL LINK("REMVTO",X,Y*(,X1,Y1,...) *)

erases all lines.

CALL LINK("TURNTO",PHI)

imperatively sets the internal angle to PHI (degrees).

CALL LINK("RECT",A,B*9,A1,B1,...)*)

draws rectangles starting from the present parameters of the cursor with the sequence A -> B -> A ->B.

CALL LINK("CLRECT",A,B*(,A1,B1,...)*)

erases the rectangles.

CALL LINK("CIRCLE",X,Y,R*(,X1,Y1,R1,...)*)

draws a circle with the center-point X,Y and the radius R.

CALL LINK("CLCRCL",X,Y,R*(,X1,Y1,R1,...)*)

erases the circles.

CALL LINK("ARCUS",X,Y,R,PHI,DPHI*(,X1,Y1,R1,PHI1,DPHI1,...)*)

draws arcs with the center-point X,Y the radius R the starting angle PHI and the arc-angle DPHI.

CALL LINK("CLARCS",X,Y,R,PHI,DPHI*(,X1,Y11,R1,PHI1,DPHI1,...)*)

erases all arc-pixels.

CALL LINK("ELLIPS",X,Y,A,B*(,X1,Y1,A1,B1,...)*)

draws ellipses with the axis center-point X,Y, of the big semi-axis A, of the small semi-axis B and the inclination PHI of the big semi-axis.

CALL LINK("CLLIPS",X,Y,A,B*(,X1,Y1,A1,B1,...)*)

erases all ellipses.

CALL LINK("VALUES",X,Y,PHI,FG,BG)

returns the present internal parameters to the variable-list, whereby X means the cursor-column, Y the cursor-row, PHI the cursor-angle, FG the fore- and BG the back-ground color.

CALL LINK("AXIS",X,LENXL,LENXR,DELTAX,Y,CLENYU,LENYD,DELTAY)

draws a system of co-ordinates with the center-point X, Y, the left X-semi-axis LENXL, the right X-semi-axis LENXR, the pitch of the X-grid DELTAX, the upper Y-semi-axis LENYU, the bottom Y-semi-axis LENDY and the pitch of the Y-grid DELTAY.

CALL LINK("HSTDIA",X,Y,WIDTH,HEIGTH,DEPTH)

draws a block diagram with the co-ordinates of the left bottom corner of the block X,Y, the width WIDTH, the height HEIGTH and the depth DEPTH.

CALL LINK("CRCDIA",X,Y,RADIUS,PHI,DPHI*(,X1,Y1,RADIUS1,PHI1,DPHI,...)*)

draws a circular diagram with the co-ordinates of the circular segment center-point X,Y, the radius RADIUS, the starting angle PHI and the finishing angle of the circular segment DPHI.

CALL LINK("WRITE",Z,S,STRING*(,Z1,S1,STRING1,...)*)

enables the mixing of graphic and text in the graphic table and writes a string (STRING) at position row (Z) and column (S) of the graphic table.

CALL LINK("DSPLAY",Z,S,SIZE,VAR$)

sets SIZE characters of the string VAR$ at position (Z,S).

CALL LINK("GSAVE","FILENAME")

saves the color of the graphic, the position of the graphic window and the graphic parameters on floppy disc in the "MEMORY-IMAGE" format.

CALL LINK("GLOAD","FILENAME"

loads a graphic called "FILENAME" from floppy disc into the VDP-RAM.

CALL LINK("DHCOPY",DESC,POP$,PADJ$)

is available only for the EPSON FX-80 printer and produces a screen-dump in the "DOWN LOAD CHARACTERSET MODE", whereby the proportional data for characters are input with DESC, the printer options with POPT$ and the printer adjustment with PADJ$.

CALL LINK("BHCOPY",MOD,POPT$,PADJ$)

generates screen-excerpts through the EPSON FX-80 or RX-80 printer in the "BIT IMAGE MODE", whereby MOD means the graphics mode of the printer, POPT$ the printer options and PADJ$ the printer adjustment.

CALL LINK("BYEBYE")

cancels the graphic mode, loads the standard character-sets and establishes the BASIC or XBASIC mode.

6.) FINAL NOTE

Congratulations!!!

You have worked through this manual. Now you can start developing your own graphic programs.

With AMERISOFT EXPANDED GRAFIC BASIC, this is simple and great fun.

We hope you understand that certain features cannot be used with AMERISOFT GRAFIC simultaneously. Not everything is possible at the same time.

Since with BASIC or XBASIC the program and the character table are in the same storage range, there are minor limitations in the application of this graphic.

It is now up to you to achieve the best results and so experience the full power of AMERISOFT EXPANDED GRAFIC BASIC, by using good programs and taking advantage of the many possibilities offered.

7.) AMERISOFT—CONDITIONS OF WARRANTY

In connection with these programs and text materials, AMERISOFT grants neither an explicit nor an implicit warranty.

In no case can AMERISOFT be held liable for additinal costs or damage in connection with the purchase or use of these programs or texts.

The sole and exclusive responsibility of AMERISOFT does not exceed the cost of purchasing the programs, whatever action be taken.

AMERISOFT does not guarantee the total absence of faults in these programs, but only the good quality of the product and the absence of material defects when handled professionally.

This program is warranted for 90 days from date of purchase.

AMERISOFT INTERNATIONAL
P.O. Box 2127
Acworth, GA 30101

NOTES

Amerisoft

LOADING PROCEDURES

FOR EXTENDED BASIC
INSERT DISK IN DRIVE ONE
SELECT EXTENDED BASIC
AFTER PROGRAM LOADS
TYPE CALL FILES (9)
NEW.

TO RUN A SAMPLE PROGRAM
TYPE OLD DSK1. UNIVDEMO
TYPE RUN

NOTE----- IF YOU HAVE NO PRINTER CONNECTED TO YOUR MACHINE IT CANNOT PRINT THE PICTURE.

EDITOR ASSEMBLER
INSERT DISK IN DRIVE ONE
SELECT TI BASIC
TYPE IN OLD DSK1. XGBSCLOAD
RUN
CALL FILES (9)
NEW
FOR A DEMO TYPE
OLD DSK1.XGBSCDEMO
RUN

FOR MINI MEMORY
INSERT DISK IN DRIVE ONE
SELECT TI BASIC
TYPE IN OLD DSK1.XGBSCLOAD1 OR OLD DSK1.XGBSCLOAD2
RUN CALL FILES (9) NEW
FOR A DEMO
TYPE IN OLD DSK1.MGBSCDEMO
RUN