TEXAS INSTRUMENTS
TI 99/4 HOME COMPUTER

CREATE ORIGINAL AND ATTRACTIVE ADVERTISING WITH YOUR TI 99/4 AND

ADVERTISER

A NOVEL AND FUN WAY TO MAKE YOUR PROGRAMMING LIFE EASIER AND SPEED UP YOUR BASIC PROGRAMS BY ADDING 39 NEW ROUTINES TO YOUR SUBPROGRAM LIBRARY

Copyright © 1981 Texas Instruments Incorporated

Program and data base contents
Copyright © 1981 Texas Instruments Incorporated

IMPORTANT NOTICE REGARDING
SUBPROGRAMS AND MANUAL

The following should be read and understood before purchasing and/or using the ADVERTISER.

TEXAS INSTRUMENTS does not warrant that the subprograms contined in the ADVERTISER or the accompanying manual will meet the specific requirements of the consumer, or that the subprograms and manual will be free from error. The consumer assumes complete responsibility for any decision made or actions taken based on information obtained using these subprograms or manual. Any statements made concerning the utility of TI's subprograms and manual are not to be construed as express or implied warranties.

TEXAS INSTRUMENTS MAKES NO WARRANTY, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THESE SUBPROGRAMS AND MANUAL OR ANY PROGRAMS DERIVED THEREFROM AND MAKES SUCH MATERIALS AVAILABLE SOLELY ON AN AS-IS BASIS.

IN NO EVENT SHALL TEXAS INSTRUMENTS BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF THE PURCHASE OR USE OF THESE SUBPROGRAMS OR MANUAL, AND THE SOLE AND EXCLUSIVE LIABILITY OF TEXAS INSTRUMENTS, REGARDLESS OF THE FORM OF ACTION, SHALL NOT EXCEED THE PURCHASE PRICE OF THIS MODULE. MOREOVER, TEXAS INSTRUMENTS SHALL NOT BE LIABLE FOR ANY CLAIM OF ANY KIND WHATSOEVER AGAINST THE USER OF THESE SUBPROGRAMS OR MATERIALS BY ANY OTHER PARTY.

SECTION 1

CONTENTS AND INTRODUCTION

SECTION I : INTRODUCTION

What is ADVERTISER? 1-3

How to use this MANUAL 1-4

How to set up your system 1-5

SECTION II : OVERVIEW OF ADVERTISER

Playing MUSIC 2-1

Delay routines PAUSE and ENTER 2-2

Introductory screens S 2-2

Display of text

- the PATTERN GENERATOR AREA 2-3
- Display D using different character sets .. 2-4
- how to use the double size letters 2-5

Sample Advertising title screen 2-6

Standard routines

- setting up a MENU 2-6
- LISTing items 2-7
- SCROLLing the screen 2-7

How to use graphics with BASIC 2-8

How to use special graphics with ADVERTISER.. 2-9

Storing graphics 2-11

CONTENTS AND INTRODUCTION

SECTION III : SAMPLE PROGRAMS

PGA 3-1

A program which displays the current graphics stored in the PATTERN GENERATOR AREA

SECTION IV : REFERENCE SECTION

ACCEPT 4-2

AD 4-4

BOXS, BOXD, BOXL 4-5

CHARSET 4-7

COLORC 4-9

COLORG 4-11

DEFS, DEFD, DEFL 4-13

DOUBLE 4-15

D 4-16

DX2 4-18

ENTER 4-20

FLASH 4-21

FRAMES, FRAMED, FRAMEL 4-23

GRAPSET 4-25

G 4-27

LIST 4-29

LOADS, LOADD, LOADL 4-31

LOGO 4-32

MENU 4-33

MUSIC 4-35

PAUSE 4-36

PREDEFS, PREDEFD, PREDEFL 4-37

S 4-39

SCROLL 4-41

SINGLE 4-42

TEXT 4-43

APPENDIX I : ERROR MESSAGES 4-45

APPENDIX II : ASCII CHARACTER CODES 4-46

APPENDIX III : GRAPHICS ORGANISATION CHART ..4-47

APPENDIX IV : GENERALLY USED PARAMETERS 4-48

APPENDIX V : ADVERTISER REFERENCE CARD 4-49

APPENDIX VI : ADVERTISER PROGRAMMING MASKS ..4-52

WHAT IS ADVERTISER?

This module contains 36 subprograms which you can use in addition to all those available when using normal BASIC. With the ADVERTISER module plugged into your machine, these are available as soon as you have selected 1 for TI BASIC from the main menu. You will not see ADVERTISER appear on the main menu because the module only contains subroutines which have to be called from normal BASIC program.

These subroutines were selected to help you create easily and quickly, eye and ear catching computer routines to run on your TI 99/4 or TI 99/4A. You can produce unique and adaptable advertising routines with ADVERTISER, the world's first interactive advertising medium; teach people about your product with simple quizzes or indeed find out about them by storing their responses when they run your program.

HOW TO USE THIS MANUAL

This manual assumes that you are already experienced in programming with TI BASIC. Statements, commands and functions that are the same as in TI BASIC are only discussed briefly here, and you should refer to the USERS REFERENCE GUIDE that came with your home computer for full details if necessary.

The additional features or ADVERTISER are explained in detail and illustrated with examples and sample programs. To get the maximum use from ADVERTISER, read this manual carefully entering and running the sample programs to see how they work. For each function, there is a full description in SECTION IV, the reference section, and you should refer to this if you are really perplexed by any of the examples.

Most of the functions in ADVERTISER can be programmed in TI BASIC; the advantages of putting them in subroutines in a module is that they execute much faster than BASIC, they take only a little time to program, and they save space in the program storage area RAM. But as the double size character set is automatically loaded into RAM, you will find that the maximum size of an ADVERTISER program is much smaller than a TI BASIC program.

The remainder of this section shows you how to set up your equipment and prepare to write an ADVERTISER program. Section II gives an overview of the abilities of ADVERTISER and contains 20 sample programs for you to try. To help you design complex graphics two programs have been designed and these you will find in SECTION III. The reference section SECTION IV contains detailed descriptions of each new function in alphabetical order together with more sample programs to illustrate their use. The APPENDICES contain much useful information on error messages, ASCII character codes, masks for programming, an ADVERTISER REFERENCE CARD (to be pulled out and used in conjunction with the BASIC reference card.

HOW TO SET UP YOUR SYSTEM

USING THE SOLID STATE SOFTWARE MODULE

3. Press the 1 key to select TI BASIC.

CONTENTS AND INTRODUCTION

4. If you are using a disk memory system then enter:

```
CALL FILES(9)     then press ENTER
NEW               then press ENTER
```

5. You are now ready to write an ADVERTISER program.

6. For every program, you must start with the initialisation function, CALL AD. Only REM statements are allowed to come before this function.

Program 1

```
>100 CALL AD
>110 CALL S(3,20,1)
>120 STOP
RUN
```

This should give you a flashing lights sequence terminated after 20 seconds.
If not, one of two error messages will appear

MESSAGE	WHAT TO DO?
INITIALISATION ERROR	Your ADVERTISER module is correctly plugged in but you have made a mistake in the initialisation procedure. Did you remember the CALL FILES instruction? Restart the initialisation.
BAD NAME	The BASIC is not recognising ADVERTISER Switch off your machine and plug in your module again, then restart the initialisation.

PROGRAM PROTECTION

Note that any user of an ADVERTISER program will have to have the ADVERTISER module plugged into his system, before he can RUN the program.

SECTION 2

OVERVIEW OF ADVERTISER

Playing MUSIC

Eleven predefined tunes have been stored in the MUSIC subroutine which can be played with just one instruction.

Program 2

```
>100 CALL AD
>110 CALL MUSIC(7)
>120 STOP
RUN
```

Notice that the music continues after the ** DONE ** message shows that the program has stopped. Once you have started a tune, it will continue and the computer can continue to do other things.

Program 3

```
>100 CALL AD
>110 CALL MUSIC(7)
>120 FOR I=1 TO 50
>130 PRINT I
>140 NEXT I
>150 STOP
RUN
```

The computer continues printing the number with the tune in the background.

Program 4

```
>100 CALL AD
>110 PRINT "TUNE"
>120 FOR I=1 TO 12
>130 PRINT TAB(15);I
>140 CALL MUSIC(I)
>150 CALL PAUSE(5)
>160 NEXT I
>170 STOP
```

This program plays all the tunes starting with the simple beep and continuing with jingles.

Of course you can also write music using the TI BASIC CALL SOUND subprogram. (See USERS REFERENCE GUIDE).

Delay routines PAUSE AND ENTER

Look again at program 4 and you will see that another ADVERTISER function was used here, PAUSE. To delay a program you can also use the function ENTER, which stops program execution until the user presses the ENTER key or until a timeout is taken. The length of the timeout is variable by the programmer.

Program 5

```
>100 CALL AD              Run and input ENTER when requested
>110 PRINT "ENTER DEMONSTRATION"
>120 PRINT
>130 CALL ENTER(4)
>RUN
                          Now Run the program again but do
                          not press ENTER. The program will
                          take the timeout after 4 seconds.
```

Introductory screens S

This is one of the most useful functions as it contains a mixture of several introductory or distracter routines, that can be used at any stage of a program. Rerun Program 1 if you have forgotten the flashing lights sequence.

The creative possibilities with these routines are very extensive but you will need to understand the graphics functions before you can fully appreciate them. Just to whet your appetite, try the following program which uses a special ADVERTISER graphic called a LOGO.

Program 6

```
>100 CALL AD
>110 CALL S(3,20,8)      The curtain drops using the
>120 CALL PAUSE(3)       default LOGO definition
>130 CALL PREDEFL(3)     One of the LOGO designs stored
>140 CALL MUSIC(7)       is selected, a jingle selected
>150 CALL S(3,20,8)      Now drop the curtain again.
>160 CALL PAUSE(3)
>170 CALL MUSIC(8)       A longer jingle is better
>180 CALL SCREEN(10)     and a change of colour
>190 CALL S(3,20,8)      And  try again. Great!
>200 CALL COLORG(10,10,13)  And don''t forget you can
>210 CALL PAUSE(3)       colour change for effect.
>220 STOP
RUN
```

Display of text

PATTERN GENERATOR AREA

The area in the computer memory RAM in which the definition of the patterns on the screen is called the PATTERN GENERATOR AREA.

In ADVERTISER for convenience, this area is divided into two halves

- the TEXT area
- the GRAPHICS area

These descriptions are only for convenience and show that normally there will be an ASCII character set in the TEXT area, (which will be used to display text); and the graphics will be defined in the GRAPHICS area. However, it is always possible to have graphics in the TEXT area and vice versa. For example, you may want to use two ASCII character sets with different colour combinations. The easiest way to do this is to load one set in the TEXT area, and the other in the GRAPHICS area.

	SETS	ASCII CHARACTER CODES
TEXT AREA	1 - 8	32 - 95
GRAPHICS AREA	9 - 16	96 - 159

Display D using different character sets

Redefining any character code can be done with the CALL CHAR(CODE,HEX$) function available with TI BASIC.

With aaa you can redefine the whole of the TEXT area, or the whole of the GRAPHICS area, by using the functions CHARSET OR GRAPSET which can load any one of six complete sets into the TEXT or GRAPHICS area respectively.

```
Program 7                Run the program and input
>100 CALL AD             any key to see the next set.
>110 PRINT "THE BEST IN THE WORLD"
>120 FOR C=1 TO 6
>130 CALL CHARSET(C)
>140 INPUT A$
>150 NEXT C
>160 STOP
RUN
```

All OK for the first three character sets but what happended when C=4. Look carefully. Can you see that only the top half of double size letters are displayed. For this set the bottom half of the characters has to be loaded into the GRAPHICS area. What happened for C=5 or 6? These two sets are graphics sets and so where letters were previously defined you now have graphics.

The PRINT statement is not very practical for many applications as it is not possible to format your output on the screen, so a function D is available which allows you to specify where on the screen you want to put your text, whether the previous text should be cleared, whether you would like a little ripple tune and a chord to emphasise your text etc.

```
Program 8

>100 CALL AD
>110 CALL D(8,4,5,24,"THE BEST IN THE WORLD")
>120 INPUT A$
>130 STOP
RUN
```

This displays the string at (4,5) with a chord played at the end of a ripple display. There are many options with this function which you need to check in the reference section.

HOW TO USE DOUBLE SIZE LETTERS

Let's do the same as in Program 8 but with double letters.

Program 9

```
>100 CALL AD              Run and input ENTER when requested
>110 CALL CHARSET(4)
>120 CALL DX2(8,4,5,24,"THE BEST IN THE WORLD")
>130 INPUT A$
>140 STOP
>RUN
```

Bizarre but lovely. There are the double size letters but there is a slight problem with colours, which are still defined as if the graphics set were loaded. Notice that you must remember to load the double size letters. Try the program again without line 110. Oops!.

Very often it would be convenient to be able to use both double-size letters and graphics at the same time. If you load the complete ASCII double size set then this is not possible. If however you can restrict the number of characters you need to 32, then the function CALL TEXT can be used. This only uses the TEXT area and leaves the GRAPHICS area free for you to use for graphics.

Program 10

```
>100 CALL AD
>110 CALL TEXT(4,5,"THE BEST IN THE WORLD")
>120 INPUT A$
>130 STOP
RUN
```

Sample Advertising Title Screen

Now run the following program which uses what you have learnt up to now to do a title screen.

Program 11

```
>100 CALL AD
>110 CALL PREDEFL(3)
>120 CALL SCREEN(10)
>130 CALL MUSIC(8)
>140 CALL S(3,20,8)
>150 CALL COLORG(10,10,13)
>160 CALL D(1,3,5,24," ")
>170 CALL D(1,5,5,24," ")
>180 CALL D(8,4,5,24," THE BEST IN THE WORLD")
>190 CALL PAUSE(3)
>200 STOP
RUN
```

Standard Routines

Setting up a MENU

One of the big advantages of ADVERTISER is that it makes programs which are run on a computer, a machine which can interact with the user. Unlike a video-machine, the programs can be tailor-made to the user. For example, if you would like to present a different range of products to each age group. Try the following program which can help you make this branch in the program.

Program 12

```
>100 CALL AD
>110 CALL D(1,2,4,24,"WHAT IS YOUR AGE")
>120 TIME=20
>130 ROW=7
>140 COLUMN=4
>150 CALL MENU(ANSWER,TIME,ROW,COLUMN,"UNDER 16",
"17-25","26-39","40-54","OVER 55")
>160 PRINT "YOUR RESPONSE WAS ";A$
>170 STOP
RUN
```

The MENU function sets up the menu, validates the user's input before returning control to the BASIC program.

LISTing items

Do you need to list out the good points of your product in an eye and ear-catching fashion? Then use the following.

Program 13

```
>100 CALL AD              Run and input ENTER when requested
>110 READ L$
>120 CALL D(8,3,4,24,"YOU NEED ADVERTISER FOR")
>130 I=I+1
>140 IF I<5 THEN 110
>150 CALL LIST(1,4,4,42,L$(0),L$(1),
 L$(2),L$(3),L$(4))
>160 STOP
>170 DATA PREDEFINED MUSIC AND #GRAPHICS
>180 DATA STANDARD ROUTINES #MENU
>190 DATA SIX CHARACTER SETS
>200 DATA SPECIAL EFFECTS ROUTINES
>210 DATA AND MUCH MORE
>RUN
```

SCROLLing the screen

Maybe you need to move the bottom of the screen up some lines. Use the SCROLL function to move the screen image. Try adding these two lines to program 13.

Program 14

```
>152 CALL SCROLL(17)
>154 INPUT A$
```

How to use graphics with BASIC

Before considering the extra abilities of ADVERTISER, let's review the possibilites available with TI BASIC.

Refer to the diagram of the PATTERN GENERATOR AREA again. Each of the ASCII codes from 32 - 159 is defined in the PATTERN GENERATOR AREA. Each of them can be redefined with CALL CHAR(CODE,HEX$) Each of them can be displayed using PRINT, CALL HCHAR or CALL VCHAR.

With ADVERTISER it is also possible to display a character with CALL D, CALL DX2 or CALL G.

The following program shows how the character code 65 which is normally the letter A can be both displayed and redefined.

Program 15

```
>100 CALL AD                        A is displayed 4 times
>110 CALL HCHAR(8,4,65,4)           with HCHAR and 3 times
>120 CALL D(1,12,4,4,"AAA")         with D
>130 CALL PAUSE(3)
>140 CALL CHAR(65,"F0F0F0F00F0F0F0F")
>150 CALL PAUSE(4)                  Now redefined
>160 PRINT "ABCABC"                 Note that the new
>170 PRINT                          definition is
>180 CALL PAUSE(3)                  permanent
>190 STOP
RUN
```

How to use special graphics with ADVERTISER

With ADVERTISER, graphics with more than one character can be displayed. and there are three types of graphic.

A 4 X 4 Character graphics stored in SETS 9 and 10
with the following Character codes.

```
LOGO      +---+---+---+---+
          |   |   |   |   |      96 - 99
          +---+---+---+---+
          |   |   |   |   |      100 - 103
          +---+---+---+---+
          |   |   |   |   |      104 - 107
          +---+---+---+---+
          |   |   |   |   |      108 - 111
          +---+---+---+---+
```

A 2 x 2 Character graphic stored in SET 9
with the following Character codes

```
DOUBLE    +---+---+
          |   |   |              96 - 97
          +---+---+
          |   |   |              98 - 99
          +---+---+
```

A 1 X 1 Character graphics stored in SET 9
with the following Character code

```
SINGLE    +---+
          |   |                  96
          +---+
```

Try the following program which will display an example of a LOGO, a DOUBLE and a SINGLE and by loading the ASCII character set show clearly the character allocation from sets 9 and 10.

Program 16

```
>100 CALL AD
>110 CALL COLORG(10,2,12)          make it pretty
>120 CALL LOGO(2,4,2)              display LOGO
>130 CALL D(1,2,10,15,"LOGO SET 9/10")
>140 CALL PAUSE(4)
>150 CALL D(1,5,10,20,"NOW LOAD ASCII CHAR")
>160 CALL D(1,7,10,20,"IN GRAPHICS AREA")
>170 CALL GRAPSET(1)               redefine LOGO
>180 CALL LOGO(12,4,2)             and like magic ...
>190 CALL PAUSE(4)
>200 CALL D(1,12,18,10,"DOUBLE")
>210 CALL DOUBLE(12,14,2)          display DOUBLE
>220 CALL D(1,13,18,10,"SINGLE")
>230 CALL SINGLE(18,14,2)          and SINGLE also
>240 CALL PAUSE(4)
>250 STOP
RUN
```

Storing graphics

With ADVERTISER it is also possible to store one LOGO one DOUBLE and one SINGLE outside the PATTERN GENERATOR AREA. It is not possible to display these without first returning them to sets 9/10 in the PATTERN GENERATOR AREA. But storing them means that they can be quickly and easily moved back and subsequently displayed, without needing to go again through their definition routine. With most of the functions using LOGO, DOUBLE or SINGLE eg FRAMED there is an option that the stored graphic is automatically moved into the PATTERN GENERATOR AREA and thus displayed.

For example, this program first shows how a LOGO is displayed and then shows the option where the stored LOGO is automatically downloaded and displayed.

Program 17

```
>100 CALL AD            Run and input ENTER when requested
>110 CALL D(1,3,3,30,"DISPLAY LOGO FROM SET 9/10")
>120 CALL LOGO(8,10,2)
>130 CALL PAUSE(3)
>140 CALL D(1,3,3,30,"DOWNLOAD STORED LOGO")
>150 CALL D(1,5,3,30,"INTO SET 9/10 & DISPLAY")
>160 CALL PAUSE(3)
>RUN
```

Changing the stored graphics can be done either by the PREDEF routines which select one of five predefined designs, or by the user redefining each character as in BASIC.

For example, add this code onto program 17 and then Run it. Here the stored LOGO is redefined to be a little lady.

Program 13

```
>170 CALL PREDEFL(3)
>180 CALL LOGO(14,10,1)
>190 CALL D(1,3,3,30,"NOW CHANGE THE STORED")
>200 CALL D(1,5,3,30,"LOGO AND DISPLAY")
>210 CALL PAUSE(3)
```

SECTION 3

SAMPLE PROGRAM
PGA

This program displays the contents of the PATTERN GENERATOR AREA

```
>100 CALL AD
>1000 CALL CLEAR
>1010 PRINT "PGA CONTENTS"
>1020 C=32
>1030 FOR Y=1 TO 16
>1040 FOR X=1 TO 15 STEP 2
>1050 CALL HCHAR(Y,X,C,1)
>1060 CALL HCHAR(Y,20,ASC(STR$(Y))
>1070 IF LEN(STR$(Y))=1 THEN 1090
>1080 CALL HCHAR(Y,21,ASC(SEG$(STR$(Y),2,1)))
>1090 C=C+1
>1100 NEXT X
>1110 NEXT Y
>1120 CALL PAUSE(3)
RUN
```

SECTION 4

REFERENCE

This section is an alphabetical list of all the aaa commands, statements and functions with a detailed explanation of how each works. Examples and programs are included wherever necessary for clarity.

ACCEPT

FORMAT

```
ACCEPT(TIMEX,ROW,COLUMN,SIZE,CODE,VAR)
ACCEPT(TIMEX,ROW,COLUMN,SIZE,CODE,VAR$
```

DESCRIPTION

The ACCEPT subprogram suspends program execution until a key is pressed on the keyboard or the timeout is reached. The cursor can be placed at any location on the screen and editing features eg DELETE, INSERT can be used. Either a string variable or a numeric variable may be requested. Existing data can be placed into the field before input is requested with the CALL D statement.

PARAMETERS

NAME	VALUES	TYPE	COMMENT
TIMEX	1 - 99	num	The time in seconds for the timeout feature
ROW	1 - 20	num	The location on the screen for
COLUMN	1 - 25	num	the first character of the variable
SIZE		num	The no of characters in the variable
	if +ve		The field is cleared before data entry
	if -ve		No clearing done
CODE		num	Return variable to the BASIC program
	0		Valid data entered
	1		Null field ie spaces entered
	2		Timeout implemented
VAR		num	Return numeric variable for the data entered
VAR$		string	Return string variable for the data entered

TIMEX: This is used if you wish the program to continue execution after a timeout has been reached without any key being pressed. Note that it is still possible to have no timeout by testing CODE on exit and looping back to the ACCEPT statement if CODE=2. (See example A)

SIZE: If the value is positive, then the field will be cleared before the entry of data. If it is negative then the input field is not blanked. This useful feature means a default value can be placed in the field before entering the ACCEPT routine, and entered just by pressing ENTER. (See example A)

CODE: This shows that valid data has been entered, or a null field (note this is allowed for a numeric entry, or the timeout taken. Be sure to check this field because only if it is 1 will the variable contain valid data.

VARS or VAR: These variables will contain the data entered in the subprogram ifit is valid. The conventions for defining variable names is as in normal BASIC.

FUNCTION KEYS

Editing keys that are available are

	99/4	99/4A
DEL	SHIFT F	FNCT 1
INS	SHIFT G	FNCT 2
CLEAR	SHIFT C	FNCT 4
LEFT	SHIFT S	FNCT S
RIGHT	SHIFT D	FNCT D

EXAMPLES

A a routine for requesting the user whether he would like to continue or not, with no timeout allowed.

```
>100  CALL  AD
>110  CALL  CLEAR
>120  CALL  D(1,4,5,17,"CONTINUE (Y/N)? Y")
>130  CALL  A(99,4,21,-1,OK,ANS$)
>140  IF    CODE=2 THEN 130
>150  PRINT "ANSWER IS ";ANS$
>160  STOP
```

AD

FORMAT

DESCRIPTION

This initialises all the variables and storage space used by the subprograms and must be the first active instruction in any program. ie only REM statements should occur before this instruction.

ERROR MESSAGES

If this instruction not used
or incorrectly used

INITIALISATION ERROR

BOX FUNCTIONS

FORMAT

```
BOXD(TYPE,ROW,COLUMN,WIDTH,DEPTH)
BOXL(TYPE,ROW,COLUMN,WIDTH,DEPTH)
BOXS(TYPE,ROW,COLUMN,WIDTH,DEPTH)
```

DESCRIPTION

These functions display a box on the screen either using a SINGLE (S), DOUBLE (D) or LOGO (L) as the building block for the display. The patterns are taken from the graphics area in the PGA but you also have the option of loading automatically the stored SINGLE, DOUBLE and LOGO.

PARAMETERS

Name	Values	Type	Function
TYPE	1	num	use as defined
	2		load stored SHAPE,DOUBLE or LOGO
ROW	1 to 22	num	Y position for top left corner
COLUMN	1 to 30	num	X position for top left corner
WIDTH	1 to 19	num	Width of box
DEPTH	1 to 27	num	Depth of box

TYPE: Normally the box will be made from the existing definition in sets 9 and 10. However if required, there is the option (TYPE=2) to load the stored SINGLE, DOUBLE or LOGO.

LOCATIONS USED BY THE BOX ROUTINES

	SET	CHAR NO
BOXS	9	96
BOXD	9	96 - 99
BOXL	9, 10	96 - 111

EXAMPLE

This will place a box of LOGOS on the screen loading the default LOGO from storage with the top left corner at (6,5)

```
>100 CALL AD
>110 CALL BOXL(2,6,5,10,12))
>120 CALL PAUSE(3)
RUN
```

CHARSET

FORMAT

CHARSET(TYPE)

DESCRIPTION

Loads any of 6 character sets into the TEXT area ie CHAR # 32-95. Note that the previous contents of these character codes will be lost after this loading.

PARAMETERS

Name	Values	Type	Function
TYPE	1	num	standard characters 8X8 dots
	2		small characters 7x5 dots
	3		small case letters
	4		double size
	5		graphics set A
	6		graphics set B

TYPE=4: In this case the character set occupies the whole of the PATTERN GENERATOR AREA. ie the TEXT and the GRAPHICS area. BEWARE!

DEFAULT

The default option is the standard character set in 32 to 95. You are recommended to keep to this structure ie putting the character set in 32 - 95 because the TEXT, MENU, LIST functions assume this is loaded in the TEXT area and all the graphics functions ie BOX,FRAME etc use the GRAPHICS area.

EXAMPLE

This program first shows the default character set and then reloads the character area with the small case letters and displays them:

```
>100 REM LOAD CHARSET AND DISPLAY PGA
>110 CALL AD
>120 GOSUB 1000
>130 CALL CHARSET(3)
>140 GOSUB 1000
>150 STOP
>1000 REM PGA DISPLAY SUBROUTINE
>1010 CALL CLEAR
>1020 PRINT "PGA CONTENTS
>1030 Y=4
>1040 C=32
>1050 FOR K=1  TO 8
>1060 FOR X=2 TO 16 STEP 2
>1070 CALL HCHAR(Y,X,C)
>1080 C=C+1
>1090 NEXT X
>1100 CALL HCHAR(Y,20,ASC(STR$(K)))
>1110 Y=Y+2
>1120 NEXT K
>1130 CALL ENTER(10)
>1140 RETURN
```

COLORC

FORMAT

COLORC(TYPE)
COLORC(10,F,B)

DESCRIPTION

The colour table for the character area is for set 1 - 8. Each set has to have a foreground colour (F) and a background colour (B). This function loads the colour table for the character area with various combinations of colours. There are some combinations specially designed for when you are working with LOGOS ie the combinations are changed only over a double set.

PARAMETERS

Name	Values	Type	Function
TYPE	1	num	random F/B
	2		random F/B by double set
	3		all F/B same as F1/B1
	4		F black B cyan
	5		circular rotation
	6		circular rotation by double set
	7		all B as B1
	8		predefined set
	9		adjust for LOGO
	10		load all with the F/B
			specified in the function

TYPE=8: This uses the predefined set of colours (which is also the default option for the graphics area).

PREDEFINED COLOUR SET

SET	FOREGROUND	F	BACKGOUND	B
1	med green	3	light green	4
2	light blue	6	dark yellow	11
3	medium red	9	cyan	8
4	magenta.	14	light yellow	12
5	light red	10	medium green	3
6	dark blue	5	grey	15
7	gray	15	medium red	9
8	medium red	9	grey	15

TYPE=10: This is the function to use when you wish to load the whole of the character area with the same combination of F and B. The colour values are the same as those in normal BASIC.

EXAMPLES

This program first shows the default colour set for the character area and then reloads the character area with the predefined colour set and displays the the results. Then all sets are loaded with light yellow on dark blue.

```
>100 REM LOAD COLOURS THEN DISPLAY PGA
>110 CALL AD
>120 GOSUB 1000
>130 CALL COLORC(8)
>140 GOSUB 1000
>150 CALL COLORC(10,12,5)
>160 GOSUB 1000
>170 STOP
>1000 REM PGA DISPLAY SUBROUTINE
>1010 CALL CLEAR
>1020 PRINT 'PGA CONTENTS
>1030 Y=4
>1040 C=32
>1050 FOR K=1  TO 8
>1060 FOR X=2 TO 16 STEP 2
>1070 CALL HCHAR(Y,X,C)
>1080 C=C+1
>1090 NEXT X
>1100 CALL HCHAR(Y,20,ASC(STR$(K)))
>1110 Y=Y+2
>1120 NEXT K
>1130 RETURN
```

COLORO.

FORMAT

COLORO(TYPE)
COLORO(10,F,B)

DESCRIPTION

The colour table for the graphics area is for set 9 - 16. Each set has to have a foreground colour (F) and a background colour (B). This function loads the colour table for the graphics area with various combinations of colours. There are some combinations specially designed for when you are working with LOGOS ie the combinations are changed only over a double set.

PARAMETERS

Name	Values	Type	Function
TYPE	1	num	random F/B
	2		random F/B by double set
	3		all F/B as F1/B1
	4		F black B cyan
	5		circular rotation
	6		circular rotation by double set
	7		all B as B1
	8		predefined
	9		adjust for LOGO making all even F/B the same as the odd F/B preceding
	10		load all with F/B these parameters being input in the function

TYPE=9: This is the predefined set of colour which is used as the default option for the graphics area. For allocation see COLORC.

TYPE=10: This is the function to use when you wish to load the whole of the graphics area with the same combination of F and B. The colour values are the same as those in normal BASIC.

EXAMPLES

This program first shows the default colour set for the graphics area. Then all sets are loaded with light yellow on dark blue.

```
>100 REM LOAD COLORS AND DISPLAY PGA
>110 CALL AD
>120 GOSUB 1000
>130 CALL COLORG(10,12,5)
>140 GOSUB 1000
>150 STOP
>1000 REM PGA DISPLAY SUBROUTINE
>1010 CALL CLEAR
>1020 PRINT "PGA CONTENTS"
>1030 Y=4
>1040 C=96
>1050 FOR K=9  TO 16
>1060 FOR X=2 TO 16 STEP 2
>1070 CALL HCHAR(Y,X,C)
>1080 C=C+1
>1090 NEXT X
>1100 CALL HCHAR(Y,20,ASC(STR$(K)))
>1110 IF LEN(STR$(K))=1 THEN 1130
>1120 CALL HCHAR(Y,20,ASC(SEG$(STR$(K),2,1)))
>1130 Y=Y+2
>1140 NEXT K
>1150 CALL ENTER(10)
>1160 STOP
```

DEFD,DEFL,DEFS

FORMAT

```
DEFD(N1,N2,N3,N4)
DEFL(N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15,N16)
DEFS(N)
```

DESCRIPTION

These routines allow you to define the stored SINGLE,DOUBLE or LOGO using existing character codes. Note that the PATTERN GENERATOR AREA is not altered in anyway by this instruction. But you must take careful note of where you stock your definitions if you change the character code definition in any way. Use the programming masks to help with this 'housekeeping'

PARAMETERS

Name	Values	Type	Function
N1	32-159	num	character code

```
SINGLE        +----+
              : N1 :                    1 pattern
              +----+
DOUBLE        +----+----+
              : N1 : N2 :               4 patterns
              +----+----+
              : N3 : N4 :
              +----+----+
LOGO          +----+----+----+----+
              : N1 : N2 : N3 : N4 :     16 patterns
              +----+----+----+----+
              : N5 : N6 : N7 : N8 :
              +----+----+----+----+
              : N9 : N10: N11: N12:
              +----+----+----+----+
              : N13: N14: N15: N16:
              +----+----+----+----+
```

EXAMPLE

This example sets up the stored double to contain small case a on the top row and 10 on the bottom row and then displays it to make sure it is correctly defined.

```
>100 REM REDEFINE DOUBLE WITH ASCII CHARACTERS
>110 CALL AD
>120 CALL CHARSET(3)
>130 CALL DEFD(ASC("A"),ASC("A"),ASC("1"),ASC("0"))
>140 CALL DOUBLE(1,5,8,1)
>150 CALL PAUSE(4)
```

DOUBLE

FORMAT

DOUBLE(TYPE,ROW,COLUMN)

DESCRIPTION

This displays a DOUBLE (2x2 patterns) on the screen at (ROW=Y,COLUMN=X) with the option of loading the stored DOUBLE.

PARAMETERS

Name	Values	Type	Function
ROW	1 to 22	num	ROW for top left corner
COLUMN	1 to 26	num	COLUMN for top left corner
TYPE	1	num	Load stored DOUBLE
	2		Use existing definition

EXAMPLE

This displays the default double character on the screen at(5,8)

```
>100 CALL AD                    This displays the star
>110 CALL DOUBLE(5,8,1)         at (Y=5,X=8)
>120 CALL PAUSE(3)
RUN
```

DISPLAY D

FORMAT

D (TYPE,ROW,COLUMN,WIDTH,I$)
D (TYPE,ROW,COLUMN,WIDTH,I)

DESCRIPTION

Displays strings or numeric variables at a specified location on the screen. The routine assumes that the standard ASCII set is loaded in the character area of the PGA. By loading other definitions you can make some very bizarre effects with this function.

PARAMETERS

Name	Values	Type	Function
TYPE	1	num	complete phrase displayed
	2		phrase displayed with chord
	3		phrase flashing for 5 s
	4		phrase flashing until input
	5		letters displayed in ripple
	6		letters individually
	7		letters individually
	8		letters ripple & chord
ROW	1 to 20	num	Y posn for first char
COLUMN	1 to 26	num	X posn for first char
WIDTH		num	width of field
	if > 0		field cleared before display
	if < 0		field not cleared before display
	WIDTH <		field truncated at length W
	LEN(I$)		
			field truncated on screen overflow
I$		a/n	string expression
I		num	numeric expression

REMARKS

EXAMPLE

```
>100 CALL AD
>110 L$='HELLO'
```

```
>120 CALL DC(1,4,3,LEN(L$),L$)
>130 CALL PAUSE(3)
```

Y= \ X=	2	3	4	5	6	7	8	9	10
3									
4		72	69	76	76	79			
5									
6									

CHARACTER CODES DISPLAYED

DX2

FORMAT

DX2(TYPE,ROW,COLUMN,WIDTH,I$)
DX2(TYPE,ROW,COLUMN,WIDTH,I)

DESCRIPTION

Displays strings or numeric variables at a specified location on the screen assuming that the double letter character set is loaded ie loads the top half of the letter from the character area and the bottom half from the graphics area.

PARAMETERS

Name	Values	Type	Function
TYPE	1	num	complete phrase displayed
	2		phrase displayed with chord
	3		phrase flashing for 5 s
	4		phrase flashing until input
	5		letters displayed in ripple
	6		letters individually
	7		letters individually
	8		letters ripple & chord
ROW	1 to 20	num	Y posn for first char
COLUMN	1 to 26	num	X posn for first char
WIDTH		num	width of field
	> 0		field cleared before display
	< 0		field not cleared before display
	WIDTH <		field truncated at length W
	LEN(I$)		
			field truncated on screen overflow
I$		a/n	string expression
I		num	numeric expression

EXAMPLE

```
>100 CALL AD
>110 L$='HELLO'
>120 CALL CHARSET(4)
>130 CALL DX2(1,4,3,LEN(L$),L$)
>140 CALL PAUSE(3)
```

X= / Y=	2	3	4	5	6	7	8	9	10
3									
4		72	69	76	76	79			
5		136	133	140	140	143			
6									

CHARACTER DISPLAYED

ENTER

FORMAT

ENTER(TIMEX)

DESCRIPTION

Displays WEITER MIT "ENTER" at Y=24 Awaits USER INPUT and verifies it unless the time exceeds the timeout specified in the parameter TIMEX. For ENTER normal exit to BASIC. For CLEAR (normally a breakpoint exit from BASIC) the program returns to the BASIC program. All other invalid entries cause a bloop and continue waiting for a response.

PARAMETERS

Name	Values	Type	Function
TIMEX	1-99	num	no of secs for timeout exit

FLASH

FORMAT

FLASH(RATE,E,EV,AREA)

DESCRIPTION

Makes the characters or graphics or both on the screen flash at various rates. Exit from the routine is either after user input or timer or both.

PARAMETERS

Name	Values	Type	Function
RATE	1 to 5	num	time between flashes
			1 the most rapid
E	1	num	exit user presses ENTER or FIRE
	2		timer operation
	3		both options
EV	1 to 120	num	time in seconds for exit
AREA	1	num	character area sets 1 to 8
	2		graphics area sets 9 to 16
	3		all PGA sets 1 to 16

EXAMPLE

This displays characters and graphics and shows the action of the FLASH command.

```
>100 CALL AD
>110 C$='CHARACTERS'
>120 G$='GRAPHICS
>130 CALL CLEAR
>140 CALL D(1,4,5,10,SEG$(C$,7,10)
>150 CALL G(1,10,5,10,SEG$(G$,7,10)
>160 CALL FLASH(3,3,5,L+1)
>170 L=L+1
>180 ON L GOTO 190,210,230
>190 PRINT C$
>200 GOTO 260
>210 PRINT G$
>220 GOTO 260
>230 PRINT C$
>240 PRINT
>250 PRINT G$
>260 PRINT
>270 CALL ENTER(10)
>270 GOTO 130
```

FRAMED.L.S

FORMAT

```
FRAMED(E,EV,TYPE,ROW,COLUMN,WIDTH,DEPTH)
FRAMEL(E,EV,TYPE,ROW,COLUMN,WIDTH,DEPTH)
FRAMES(E,EV,TYPE,ROW,COLUMN,WIDTH,DEPTH)
```

DESCRIPTION

Using a SINGLE, DOUBLE OR LOGO as the building block, this draws a colour frame round an area with colour bands changing randomly. Exit from the routine can be either from user intervention or from a timer set or both. Top left corner of the frame is at (ROW=Y,COLUMN=X) and the frame will just be truncated in the case of an overflow.

PARAMETERS

Name	Values	Type	Function
S	1	num	if ENTER or FIRE button pressed
	2		if timer operation
	3		if both input and timer to be used
SV	1 to 120		number of seconds to wait
TYPE	1	num	selected randomly
	2		selected sequentially
	3		whole band then
			value changed sequentially
ROW	1 to 20	num	Y position for top left
COLUMN	1 to 26	num	X position for top right
WIDTH	1 to 32	num	no of patterns horizontally
			X+NX <= 28
DEPTH	1 to 24	num	no of patterns vertically
			Y+NY <= 24

EXAMPLE

This draws a frame of DOUBLES from which exit is either after user input or a timeout of 10 seconds.

```
>100 CALL AD
>110 CALL CLEAR
>120 CALL FRAMED(3,10,3,5,6,10,10)
>130 GOTO 130
```

GRAPSET.

FORMAT

GRAPSET(TYPE)

DESCRIPTION

Loads any of 5 graphics sets into the GRAPHICS area ie CHAR # 96 159. Note that the previous contents of these character codes will be lost after this loading.

PARAMETERS

Name	Values	Type	Function
TYPE	1	num	standard characters 8X8 dots
	2		small characters 7x5 dots
	3		small case letters
	4		not available
	5		graphics set A
	6		graphics set B

DEFAULT

The default option is the graphics set At in 96 to 159. You are recommended to keep to this structure ie putting the character set in 32 - 95 and the graphics in 96 - 159 because the TEXT, MENU, LIST functions assume this is loaded in the TEXT area and all the graphics functions ie BOX,FRAME etc use the GRAPHICS area.

EXAMPLE

This program first shows the default graphics set and then reloads the graphic area with the standard letters and displays them.

```
>100 REM LOAD TEXT INTO GRAPHICS AREA
>110 CALL AD
>120 GOSUB 1000
>130 CALL GRAPSET(1)
>140 GOSUB 1000
>150 STOP
>1000 REM PGA DISPLAY SUBROUTINE
>1010 CALL CLEAR
>1020 PRINT 'PGA CONTENTS'
>1030 Y=4
>1040 C=96
>1050 FOR K=9  TO 16
>1060 FOR X=2 TO 16 STEP 2
>1070 CALL HCHAR(Y,X,C)
>1080 C=C+1
>1090 NEXT X
>1100 CALL HCHAR(Y,20,ASC(STR$(K)))
>1110 IF LEN(STR(K))=1 THEN 1130
>1120 CALL HCHAR(Y,20,ASC(SEG(STR$(K),2,1)))
>1130 Y=Y+2
>1140 NEXT K
>1150 CALL ENTER(10)
>1160 RETURN
```

DISPLAY G

FORMAT

```
G(TYPE,ROW,COLUMN,WIDTH,I$)
G(TYPE,ROW,COLUMN,WIDTH,I)
```

DESCRIPTION

Displays strings or numeric variables at a specified location on the screen from the graphics area. The routine assumes that the standard ASCII set is loaded in the graphics area of the PGA. By loading other definitions you can make some very bizarre effects with this function.

PARAMETERS

Name	Values	Type	Function
TYPE	1	num	complete phrase displayed
	2		phrase displayed with chord
	3		phrase flashing for 5 s
	4		phrase flashing until input
	5		letters displayed in ripple
	6		letters individually
	7		letters individually
	8		letters ripple & chord
ROW	1 to 20	num	Y posn for first char
COLUMN	1 to 26	num	X posn for first char
WIDTH		num	width of field
	if > 0		field cleared before display
	if < 0		field not cleared before display
	WIDTH <		field truncated at length W
	LEN(I$)		
			field truncated on screen overflow
I$		a/n	string expression
I		num	numeric expression

EXAMPLE

```
>100 CALL AD
>110 CALL GRAPSET(2)
>120 L$="HELLO"
>130 CALL G(1,4,3,LEN(L$),L$)
>140 CALL PAUSE(3)
```

X= / Y=	2	3	4	5	6	7	8	9	10
3									
4		136	133	140	140	143			
5									
6									

CHARACTER CODES DISPLAYED

LIST

FORMAT

LIST(TYPE,ROW,COLUMN,N,I1$,I2$.....)

DESCRIPTION

Displays each item successively on the screen, pointed to by a character and emphasised by a tune. Each successive item being displayed below with double-spacing. When the list reaches the bottom of the screen either the whole screen is scrolled up so that the next item is displayed at the bottom etc etc or the action is halted and 'WEITER MIT "ENTER"' sequence is entered, the screen is cleared and the rest of the strings are displayed in the same manner. The timeout for the PRESS ENTER routine depends on the parameter values.

PARAMETERS

Name	Values	Type	Function
TIME	1	num	scrolling on overflow
	2-99		WEITER MIT "ENTER" routin on
			overflow with timeout= TIME secs
ROW	1 to 20	num	Y posn for first pointer
COLUMN	1 to 26	num	X posn for first pointer
N	32-159	num	pattern used for pointer
I1$		a/n	display of any string expression

NOTE: Multiple Items can be entered in one LIST command up to the length of the line in BASIC.

EXAMPLE:

```
>100 REM LIST ITEMS
>110 CALL AD
>120 A$='FIRST LINE OF LISTING'
>130 B$='NEXT LINE OF LISTING'
>140 C$='NEXT LINE##WITH CONTROL CHA#RAC#TERS'
>150 CALL LIST(10,7,4,42,A$,B$,C$)
>160 CALL PAUSE(3)
```

LOADD,LOADC,LOADS

FORMAT

LOADD(TYPE)
LOADC(TYPE)
LOADS(TYPE)

DESCRIPTION

Loads stored DOUBLE,LOGO,SINGLE into part of the PATTERN GENERATOR AREA. This instruction is needed before S or FRAME or to retrieve the stored graphics into the graphics area.

PARAMETERS

Name	Values	Type	Function
TYPE	1	num	Load into set 9
	2		Load into all sets 9-16
	3		Load into last 6 sets 11-16 for FRAME routine

LOGO

FORMAT

LOGO(ROW,COLUMN,TYPE)

DESCRIPTION

Displays LOGO on the screen with top left corner at (ROW=Y,COLUMN=X) with the option of loading a stored LOGO.

PARAMETERS

Name	Values	Type	Function
ROW	1 to 20	num	Location of top right hand
COLUMN	1 to 26	num	Location of top right hand
TYPE	1	num	Load stored LOGO into 96/111
	2		Use existing definition of 96/111

EXAMPLE

This displays a LOGO on the screen at (5,8) drawn from the default stored LOGO.

```
>100 CALL AD                    This displays the star
>110 CALL LOGO(5,8,1)           at (Y=5,X=8)
>120 CALL PAUSE(3)
```

MENU

FORMAT

MENU(VAR,TIME,ROW,COLUMN,I1$,I2$.....)

DESCRIPTION

This is a complete routine to display a menu, await and verify the input containing the following features: Displays 'BITTE WAEHLEN SIE' at location (ROW,COLUMN); Displays 'IHRE WAHL' at location (22,12); Displays each item successively on the screen, number at X=COLUMN+2 Awaits input and verifies within range. If no input after Time then returns to BASIC leaving zero in the return variable. Items double spaced.

PARAMETERS

Name	Values	Type	Function
VAR			defined in program
			used to pass input back to user
			0 if timeout taken
TIME	1 to 99	num	timeout in secs
ROW	1 to 20	num	ROW posn for PRESS FOR
COLUMN	1 to 26	num	COLUMN posn for PRESS FOR
I1$	max in 48	a/n	string expressions

NOTE: Overflow of the menu onto another screen is not allowed so the number of menu items allowed is limited by the size of the screen.

EDITING CHARACTERS

The character # in a string means that a new line is taken.

EXAMPLE

This example displays the menu in the format below then awaits a user input for 20 seconds.

```
   0.........1.........2.........3.
 0 **                              **  10 CALL AD
 1 **                              **  20 A$="RED"
 2 **                              **  30 B$="BLUE"
 3 **BITTE WAEHLEN SIE             **  40 C$="GREEN"
 4 **                              **  50 D$="YELLOW"
 5 **  1   RED                     **  60 CALL MENU(R,20,3,2,A$,B$,C$,D$)
 6 **                              **
 7 **  2   BLUE                    **
 8 **                              **
 9 **  3   GREEN                   **
10 **                              **
11 **  4   YELLOW                  **
12 **                              **
13 **                              **
14 **                              **
15 **                              **
16 **                              **
17 **                              **
18 **                              **
19 **                              **
20 **                              **
21 **                              **
22 **              IHRE WAHL? #    **
23 **                              **
```

If there is any overflow for any item, the item is split at the last whole word and continued on the next line.

MUSIC

FORMAT

MUSIC(TYPE)

DESCRIPTION

Plays a predefined TUNE.

PARAMETERS

Name	Values	Type	Function
TYPE	1	num	Beep
	2		Bloop
	3		C major
	4		start
	5		end
	6		reward 1
	7		reward 2
	8		reward 3
	9		listing
	10		mistake 1
	11		mistake 2
	12		Silence

EXAMPLE

To play all the tunes

```
>100 CALL AD
>110 FOR K=1 TO 12
>120 CALL MUSIC(K)
>130 NEXT K
>140 STOP
```

PAUSE

FORMAT

PAUSE(TYPE)

DESCRIPTION

Wait for a number of seconds

PARAMETERS

Name	Values	Type	Function
TYPE	1-16	num	No of seconds

PREDEFD,PREDEFL,PREDEFS

FORMAT

```
PREDEFD(TYPE)
PREDEFL(TYPE)
PREDEFS(TYPE)
```

DESCRIPTION

Loads the stored SINGLE,DOUBLE OR LOGO with a predefined pattern

PARAMETERS

PREDEFS

Name	Values	Type	Function
TYPE	1	num	Star
	2		Diamond
	3		Harlequin
	4		Dots
	5		Spots

PREDEFD

Name	Values	Type	Function
TYPE	1	num	Star
	2		Diamond
	3		Harlequin
	4		Dots
	5		Spots

PREDEFL

Name	Values	Type	Function
TYPE	1	num	Star
	2		Diamond
	3		Lady
	4		Man
	5		Checkers

EXAMPLE

To create a display with a woman speaking

```
>100 CALL AD
>110 CALL CLEAR
>120 CALL PREL(3)
>130 FOR Y=3 TO 22)
>140 CALL HCHAR(Y,5,96,25)
>150 NEXT Y
>130 CALL LOGO(16,5,1)
>140 CALL COLORC(10,13,12)
>150 CALL TEXT(12,8," HI THERE # HOW ARE YOU TODA
>160 CALL PAUSE(3)
```

S

FORMAT

S(E,EV,TYPE)

DESCRIPTION

Clears the original screen and displays a whole screen and accompanying music, until the user presses FIRE or ENTER or timeout. There are 5 different types of routines available, and with each one there is the possibility of using the existing definition of the PGA and colors or to load stored values.

PARAMETERS

Name	Values	Type	Function
E		num	WHAT TYPE OF EXIT
	1		user press ENTER or FIRE
	2		exit on timer operation
	3		on either input or timer operation
EV		num	TIMER VALUE?
	0 to 120		number of seconds to wait
TYPE	1	num	flashing lights
	2		flashing lights
	3		LOGO stripes and colour wash
			using stored LOGO
	4		LOGO stripes and colour wash
			using stored LOGO
	5		LOGO stripes at random
	6		LOGO stripes at random with stored LOGO
	7		LOGO curtain down
	8		LOGO curtain down with stored LOGO
	9		LOGO strip up
	10		LOGO strip up with stored LOGO

TYPE: If you select to load the LOGO from the stored logo then the character codes 96 - 111 will be overwritten when the stored LOGO set is loaded into set 9,10. For the colour wash, sets 9-12 will be overwritten.

EXAMPLE

To set up a curtain of black and blue men dropping from the sky.

```
>100 CALL AD
>110 CALL COLORC(10,2,10)
>120 CALL CLEAR
>130 CALL PREDEFL(4)
>140 CALL LOADL(1)
>150 CALL COLORG(4)
>160 CALL S(3,10,7)
>170 CALL PAUSE(3)
```

FORMAT

SCROLL(TYPE)

DESCRIPTION

Scrolls the screen display vertically upward by a specified no of lines filling in any extra lines with blanks ie char 32.

PARAMETERS

Name	Values	Type	Function
TYPE	1 to 32	num	no of lines to scroll

EXAMPLE

Print some text on the screen and then scroll.

```
>100 CALL AD
>110 CALL CLEAR
>120 CALL TEXT:12,4:"LETS TEST##     SCROLL"
>130 CALL ENTER(10)
>140 CALL SCROLL(4)
>150 GOTO 130
```

This example clearly shows that the characters in the character area are redefined with the TEXT function, giving the strange effects on the WEITER MIT "ENTER" message.

TEXT

FORMAT

TEXT (ROW,COLUMN,I$)

DESCRIPTION

Displays a string on the screen, using double letter format. This routine uses the TEXT area for the double letter patterns so after return to BASIC these will still be loaded in the TEXT area.
Note that this uses the whole of the TEXT area each time it is called so that more than one CALL should not be made without clearing the screen.
A # characters in the string is recognised as a request to start a new line.

PARAMETERS

Name	Values	Type	Function
ROW	1 to 22	num	Y posn for first char
COLUMN	1 to 23	num	X posn for first char
I$		a/n	display of any string expression

ERROR MESSAGES

The maximum no of different characters allowed in one string is 31 excluding the space character. Error message is displayed if the number of characters exceeds this.

error message:
STRING OVERFLOW

SCROLL

FORMAT

SCROLL(TYPE)

DESCRIPTION

Scrolls the screen display vertically upward by a specified no of lines filling in any extra lines with blanks ie char 32.

PARAMETERS

Name	Values	Type	Function
TYPE	1 to 32	num	no of lines to scroll

EXAMPLE

Print some text on the screen and then scroll.

```
>100 CALL AD
>110 CALL CLEAR
>120 CALL TEXT(12,4,"LETS TEST##     SCROLL"
>130 CALL ENTER(10)
>140 CALL SCROLL(4)
>150 GOTO 130
```

This example clearly shows that the characters in the character area are redefined with the TEXT function, giving the strange effects on the WEITER MIT "ENTER" message.

TEXT

FORMAT

TEXT (ROW,COLUMN,IS)

DESCRIPTION

Displays a string on the screen, using double letter format. This routine uses the TEXT area for the double letter patterns so after return to BASIC these will still be loaded in the TEXT area.
Note that this uses the whole of the TEXT area each time it is called so that more than one CALL should not be made without clearing the screen.
A # characters in the string is recognised as a request to start a new line.

PARAMETERS

Name	Values	Type	Function
ROW	1 to 22	num	Y posn for first char
COLUMN	1 to 28	num	X posn for first char
IS		a/n	display of any string expression

ERROR MESSAGES

The maximum no of different characters allowed in one string is 31 excluding the space character. Error message is displayed if the number of characters exceeds this.

error message:
STRING OVERFLOW

EXAMPLE

```
EXAMPLE
100     CALL AD
110     CALL TEXT(4,3,"HELLO")
120     CALL PAUSE(8)
```

This prints HELLO on the screen at (Y=4,X=3)
And defines the TEXT area as

	SET#	DEFINITION OF CHARACTER CODES													
top half	1	32	33	H	34	E	35	L	36	O	37		38	39	
	2	40	41		42		43		44		45		46	47	
	3	48	49		50		51		52		53		54	55	
	4	56	57		58		59		60		61		62	63	
bottom half	5	64	65	H	66	E	67	L	68	O	69		70	71	
	6	72	73		74		75		76		77		78	79	
	7	80	81		82		83		84		85		86	87	
	8	88	89		90		91		92		93		94	95	

```
130     CALL TEXT(14,3,"THERE!!!")
140     GOTO 140
```

Because this will redefine the character area displayed on the screen will be THERE!!! at (Y=14,X=3) and the previous phrase redefined will be displayed as THEER

	SET#	DEFINITION OF CHARACTER CODES													
top half	1	32	33	T	34	H	35	E	36	R	37	!	38	39	
	2	40	41		42		43		44		45		46	47	
	3	48	49		50		51		52		53		54	55	
	4	56	57		58		59		60		61		62	63	
bottom half	5	64	65	T	66	H	67	E	68	R	69	!	70	71	
	6	72	73		74		75		76		77		78	79	
	7	80	81		82		83		84		85		86	87	
	8	88	89		90		91		92		93		94	95	

APPENDIX I ERROR MESSAGES

ERROR MESSAGES	comments
INCORRECT STATEMENT	
STRING/NO MISMATCH	TEXT,LIST,MENU items must be strings MENU return variable must be numeric
PARAMETER OUT OF RANGE	
INITIALISATION ERROR	CALL FILES(9) not entered or CALL AD not first instruction
STRING OVERFLOW	TEXT limited to 32 characters including the SPACE
PAGE OVERFLOW	MENU page overflow not allowed
BREAKPOINT	shift C was pressed

APPENDIX II : ASCII CHARACTER CODES

PATTERN GENERATOR AREA

TEXT AREA		GRAPHICS AREA	
Set	ASCII codes	Set	ASCII CODES
1	32-39	9	96-103
2	40-47	10	104-111
3	48-55	11	112-119
4	56-63	12	120-127
5	64-71	13	128-135
6	72-79	14	136-143
7	80-87	15	144-151
8	88-95	16	152-159

APPENDIX III : GRAPHICS STORAGE

The user can define and store 1 SINGLE (1x1 pattern)
1 DOUBLE (2x2 patterns)
1 LOGO (4x4 patterns)
The default value of all these stored graphics is a star

APPENDIX IV : GENERALLY USED PARAMETERS

E

1 user presses ENTER or FIRE
2 timeout exit
3 either input or timeout

EV

1 - 120 no of seconds before exit

ROW,COLUMN

1 - 20,1-26 coordinates of top left corner

WIDTH,DEPTH

1 - 20,1-26 no of patterns width/depth

APPENDIX V : ADVERTISER REFERENCE CARD

INITIALISATION PROCEDURE

After selecting TI BASIC

USING A DISK SYSTEM Enter CALL FILES(9)

FOR ALL USERS Then NEW
100 CALL AD

```
ACCEPT(TIMEX,ROW,COLUMN,SIZE,CODE,VAR or VAR$)   input data
return code 0 valid data 1 null entry 2 timeout
return variable VAR$ if string VAR if numeric
                                                 initialisation
BOXD(TYPE,ROW,COLUMN,WIDTH,DEPTH)                draw box doubles
BOXS(TYPE,ROW,COLUMN,WIDTH,DEPTH)                draw box singles
BOXL(TYPE,ROW,COLUMN,WIDTH,DEPTH)                draw box logos
CHARSET(TYPE)                                    load character area
TYPE 1: standard 2: small 3:small case 4:double 5:graphics A 6:graphics B
COLORC(TYPE)    COLORC(10,F,B)                   load character colors
COLORG(TYPE)    COLORG(10,F,B)                   load graphics colors
TYPE 1: random 2: random(x2) 3: as set 1 4: black on cyan 5: circular rot.
     6: circular rot(x2) 7: B as B1 8: predefined 9: for LOGO
     10: all F/B
DEFD(N1,N2,N3,N4)                                define stored DOUBLE
DEFL(N1,N2,N3,N4,......N16)                      define stored LOGO
DEFL(N)                                          define stored SINGLE
DOUBLE(TYPE,ROW,COLUMN)                          display double
TYPE 1: load stored DOUBLE 2: use existing
D(TYPE,ROW,COLUMN,WIDTH,I$ or I)                 display string,number
TYPE 1: whole phrase 2:  phrase with chord 3: phrase flashed for 5 s
     4: phrase flashed until input 5: ripple letters 6: individual letters
     7: individual letters 8: ripple with chord
DX2(TYPE,ROW,COLUMN,WIDTH,I$ or I)               display string/number
TYPE :as D                                       with double letters
ENTER(TIMEX)                                     press enter to continue
FLASH(RATE,E,EV,AREA)                            flash
RATE: 1 - 5 1 most rapid   AREA: 1 char 2 grap 3 char & grap
FRAMED(E,EV,TYPE,ROW,COLUMN,WIDTH,DEPTH)         draw frame of DOUBLE
FRAMEL(E,EV,TYPE,ROW,COLUMN,WIDTH,DEPTH)         draw frame of LOGO
FRAMES(E,EV,TYPE,ROW,COLUMN,WIDTH,DEPTH)         draw frame of LOGO
TYPE: 1 random 2 sequential 3 whole band then change
GRAPSET(TYPE)                                    load characters/graphics
TYPE: as CHARSET except TYPE <> 4                sets in graphics area
G(TYPE,ROW,COLUMN,WIDTH,I$ or I)                 display from graphics
```

```
TYPE :as D                                                  area
LIST(TYPE,ROW,COLUMN,N,I1$,I2$.....)                        listing strings
TYPE: 1 scrolling 2 - 99 press enter to continue on overflow with
                    timeout after 2 - 99 seconds
LOADD(TYPE)                                                 load DOUBLE into graphic
LOADL(TYPE)                                                 load LOGO into graphics
LOADS(TYPE)                                                 load SINGLE into graphi
LOGO(ROW,COLUMN,TYPE)                                       display LOGO
TYPE: 1: load stored LOGO  2: use existing
MENU(VAR,TIME,ROW,COLUMN,I1$,I2$.....)                      set up menu and get inp
VAR: variable name for answer or 0 if timeout taken  TIME: timeout in se
MUSIC(TYPE)                                                 play predefined tune
TYPE: 1 beep  2 bloop  3 c major scale  4 start  5 end
      6,7,8  reward  9 listing  10,11  mistake  12 silence
PAUSE(TYPE)                                                 wait for TYPE secs
TYPE 1 - 16
PREDEFD(TYPE)                                               select DOUBLE predefin
PREDEFL(TYPE)                                               select LOGO predefined
PREDEFS(TYPE)                                               select SINGLE predefine
S(E,EV,TYPE)                                                intro screen
TYPE  1,2 flashing lights 3,4 LOGO stripes  5,6 LOGO stripes at random
      7,8 LOGO curtain down 9,10 LOGO strip up
      The lower no of each pair using the existing PGA.
SCROLL(TYPE)                                                scroll up TYPE lines
TYPE 1 - 32
SINGLE(ROW,COLUMN,TYPE)                                     display SINGLE
TYPE 1 load stored SINGLE   2 use as defined
TEXT(ROW,COLUMN,I$)                                         display string(x2 size)
```

APPENDIX VI : PROGRAMMING MASKS

```
   0.........1.........2.........3.
 0 **                             **
 1 **                             **
 2 **                             **
 3 **                             **
 4 **                             **
 5 **                             **
 6 **                             **
 7 **                             **
 8 **                             **
 9 **                             **
10 **                             **
11 **                             **
12 **                             **
13 **                             **
14 **                             **
15 **                             **
16 **                             **
17 **                             **
18 **                             **
19 **                             **
20 **                             **
21 **                             **
22 **                             **
23 **                             **
   0.........1.........2.........3.
```

GRAPHICS AND CHARACTER DEFINITION

SET CHAR NO. and CONTENTS

SET#	COLR	F/B	DEFINITION OF CHARACTER CODES									
1			32	33	34	35	36	37	38	39		
2			40	41	42	43	44	45	46	47		
3			48	49	50	51	52	53	54	55		
4			56	57	58	59	60	61	62	63		
5			64	65	66	67	68	69	70	71		
6			72	73	74	75	76	77	78	79		
7			80	81	82	83	84	85	86	87		
8			88	89	90	91	92	93	94	95		
9			96	97	98	99	100	101	102	103		
10			104	105	106	107	108	109	110	110		
11			112	113	114	115	116	117	118	119		
12			120	121	122	123	124	125	126	127		
13			128	129	130	131	132	133	134	135		
14			136	137	138	139	140	141	142	143		
15			144	145	146	147	148	149	150	151		
16			152	153	154	155	156	157	158	159		

GRAPHICS AND CHARACTER DEFINITION MASK
H. NO ASCII CHARACTER SET LOADED

SET CHAR NO. and CONTENTS

SET#	COLS	F/B	DEFINITION OF CHARACTER CODES								
1			32	33	34	35	36	37	38	39	
2			40	41	42	43	44	45	46	47	
3			48	49	50	51	52	53	54	55	
4			56	57	58	59	60	61	62	63	
5			64	65	66	67	68	69	70	71	
6			72	73	74	75	76	77	78	79	
7			80	81	82	83	84	85	86	87	
8			88	89	90	91	92	93	94	95	
9			96	97	98	99	100	101	102	103	
10			104	105	106	107	108	109	110	110	
11			112	113	114	115	116	117	118	119	
12			120	121	122	123	124	125	126	127	
13			128	129	130	131	132	133	134	135	
14			136	137	138	139	140	141	142	143	
15			144	145	146	147	148	149	150	151	
16			152	153	154	155	156	157	158	159	