

Foundations 3:

Graphics, modes and sound

The Amstrad CPC 464 Colour Personal Computer has three modes of screen display operation: Mode 0, Mode 1, and Mode 2.

When the computer is first switched on, it is automatically in Mode 1.

To understand the different modes, switch on the computer and press the number **1** key. Hold it down until two lines are full of number 1 's. If you now count the number of 1 's on a line, you will see that there are 40. This means that in mode **1**, there are 40 columns. Press [ENTER] - you will get a **S y n t a x e r r o r** message, but don't worry, this is just a quick way of getting back to the **R e a d y** message that tells you the computer is waiting for your next instruction.

Now type in: **mod e 0** [ENTER]

You will see the characters on the screen are now larger. Press the number **1** key again and hold it down until two lines are full of 1 's. If you count the number of 1 's on a line, you will see there are 20. This means that in mode 0, there are 20 columns. Press [ENTER] again.

Now type in: **mod e 2** [ENTER]

You will see that this is the smallest mode, and if you type in a row of 1 's, you will count 80. This means that in mode 2 there are 80 columns. To recap:

Mode 0 □ 20 columns

Mode 1 □ 40 columns

Mode 2 □ 80 columns

Finally, press [ENTER] once more.

COLOURS

There is a choice of 27 colours. These are shown on a green monitor (GT 64) as various shades of green. If you purchased the GT 64 monitor, you can buy the **AMSTRAD MP1** Modulator/Power supply in order to use the computer's colour facilities on your domestic colour T.V.

In Mode 0, up to 16 of the 27 available colours can be put on to the screen at any time.

In Mode 1, up to 4 of the 27 colours can be put on to the screen at any time.

In Mode 2, up to 2 of the 27 colours can be put on to the screen at any time.

You are able to change the colour of the **B O R D E R**, the **P A P E R** (the area where the characters can appear) or the **P E N** (the character itself), all independently of each other.

The 27 colours available are listed in Table 1, each with their I N K reference number .

MASTER COLOUR CHART

Ink Number	Colour/Ink	Ink Number	Colour/Ink ,
0	Black	14	Pastel Blue
1	Blue	15	Orange
2	Bright Blue	16	Pink
3	Red	17	Pastel Magenta
4	Magenta	18	Bright Green
5	Mauve	19	Sea Green
6	Bright Red	20	Bright Cyan
7	Purple	21	Lime Green
8	Bright Magenta	22	Pastel Green
9	Green	23	Pastel Cyan
10	Cyan	24	Bright Yellow
11	Sky Blue	25	Pastel Yellow
12	Yellow	26	Bright White
13	White		

Table one: The I N K numbers and colours

As explained earlier, when the computer is first switched on, it is in Mode 1. To return to Mode 1, type in:

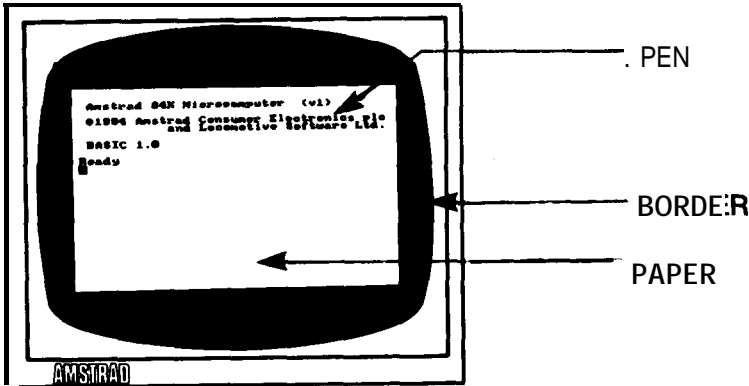
mod e 1 [ENTER]

The colours of **BORDER**, **PAPER**, and **PEN** at switch-on are:

Border: Colour number 1 (blue)

Paper (screen): Colour number 1 (blue)

Pen (characters): Colour number 24 (bright yellow)



The **BORDER** is the area surrounding the **PAPER**. (Note that when the computer is first switched on, the **BORDER** and **PAPER** are both blue). The **P A P E R** is the area of the screen inside the **B O R D E R** where characters can appear. The **P E N** is the **colour** of the characters.

To explain this further, we can associate the PEN and PAPER on the monitor screen to an actual P E N and piece of notepaper. As the colour of the I N K in a P E N can be changed, so the colour of the characters on the screen can be changed. As the colour of notepaper can be changed, so the colour of the PA P E R on the screen can also be changed.

To change the colour of the B O R D E R, type in:

`border 0 [ENTER]`

You will see the **B0** R D E R colour change from blue to black. If you refer to Table 1, you will see that 0 is black. The B O R D E R can be changed to any of these colours by typing in: `border` then the colour number required.

Now type in:

`cls [ENTER]`

to clear the screen.

To see the PA P E R colour change, type in:

`paper 2 [ENTER]`

You will see the background colour behind the word Ready change to bright cyan. Now type in:

`cls [ENTER]`

to clear the screen again to the new PA P E R colour.

To see the P E N colour change, type in:

`PEN 3 [ENTER]`

You will see that the PEN colour has changed and the word R **e** a d y is printed in bright red,

Now type in:

`cls [ENTER]`

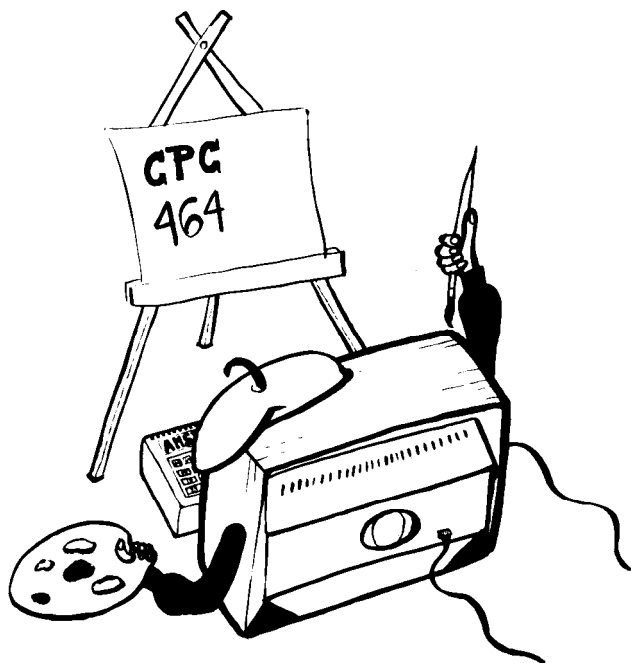
You will find it easier to understand what has happened here if you refer also to Table 2. When the computer is first switched on, the PA P E R used is number 0. If you look at Table 2, in the first column, you will see PA P E R number 0. If you now look along the same line in the Mode 1 column, you will see colour number 1.

If you now refer to the master colour chart (Table 1), you will see that number 1 is equal to blue, which is the colour of the PA P E R when the computer is first switched on.

We have just changed the PAPER to PAPER number 2. Refer to PAPER number 2 in the left column of Table 2, and you will see in the mode 1 column that this is colour number 20. Now look at Table 1 and you will see that colour 20 is bright cyan.

Ink Colour			
<i>Paper/PenNo.</i>	<i>Mbde0</i>	<i>Mbde1</i>	<i>Mbde2</i>
0	1	1	1
1	24	24	24
2	20	20	1
3	6	6	24
4	26	1	1
5	0	24	24
6	2	20	1
7	8	6	24
8	10	1	1
9	12	24	24
10	14	20	1
11	16	6	24
12	18	1	1
13	22	24	24
14	Flashing 1,24	20	1
15	Flashing 16,11	6	24

Table two: PAPER / **PEN/MODE/INK** reference



The **P E N** used when the computer is first switched on, is number 1. Look at Table 2 and you will see in the mode 1 column, that **P E N** number 1 is colour number 24. If you now refer to Table 1, you will see that colour number 24 is bright yellow, which is the colour of the characters (**P E N**) when the computer is first switched on.

We have just changed the **P E N** to number 3. Looking at Table 2, you will see that **P E N** number 3 is colour number 6 in mode 1. Now refer to Table 1, and you will see that colour 6 is bright red.

At the moment, we are using **P A P E R** number 2 and **P E N** number **3**. We can now go on to change the colour of these. We do this by using the **I N K** command. The **I N K** command has two numbers, the first is the number of the **P E N** or **P A P E R** to be changed, the second is the colour that the **P E N** or **P A P E R** is to be changed to. Look at Table 1 for the colour numbers. As an example, we will show how to change the colour of **P A P E R 2** to black and the colour of **P E N 3** to bright white.

In Table 1, you will see that the colour number for black is 0, and the colour number for bright white is 26.

Now type in:

```
ink 2,0 [ENTER]
```

(As explained, 2 is the current **P A P E R** number, and 0 is black.)

Now type in:

```
ink 3,26 [ENTER]
```

(3 is the current **P E N** number and 26 is bright white)

Now fully reset the computer by pressing [**CTRL**] [**SHIFT**] and [**ESC**] keys.

As explained earlier, when the machine is first switched on, or reset using [**CTRL**] [**SHIFT**] [**ESC**], the **P A P E R** number is 0 and the **P E N** number is 1. The colour of the **P A P E R** is 1 (blue) and the colour of the **P E N** is 24, (bright yellow). These would be typed in as **ink 0,1** for the **P A P E R**, and **ink 1, 24** for the **P E N**. To change these immediately to white characters (**P E N**) on a black background (**P A P E R**), type in:

```
ink 0,0 [ENTER]
```

Then type in:

```
ink 1,26 [ENTER]
```

FLASHING COLOURS

It is possible to make the colour of the characters flash between one colour and another. This can be achieved by adding an extra colour number to the I N K command of the P E N .

To see the characters on the screen flashing between bright white and bright red, type in:

```
ink 1,26,6 [ENTER]
```

In this case, 1 is the P E N number, while 26 is the colour bright white, and 6 is the alternate colour, bright red.

It is also possible to make the colour of the P A P E R behind the characters flash between one colour and another. This can be achieved by adding an extra colour number to the I N K command for the P A P E R .

To see the PAPER flashing between green and bright yellow, behind the characters, type in:

```
ink 0,9,24 [ENTER]
```

In this case 0 is the PAPER number. while 9 is the colour green, and 24 is the alternate colour, bright yellow.

Now reset the computer, [CTRL] [SHIFT] [ESC]

Note that in mode 0, two of the P E Ns (numbers 14 and 15), together with two of the P A P E Rs (numbers 14 and 15) are default flashing colours. In other words, it is not necessary to add an extra number to the I N K command.

type in the following:

```
mode 0 [ENTER]  
pe n 15 [ENTER]
```

on the screen you will see the word R e a d y flashing between sky blue and pink.

Now type in:

```
paper 14 [ENTER]  
c 1 s [ENTER]
```

You will now see that in addition to the word R e a d y flashing between sky blue and pink, the background P A P E R is also flashing between yellow and blue.

It is possible to change these default flashing colours by typing in a new I N K command for the P E N or P A P E R. To change the colour of the P E N to flashing black and bright white, type in:

```
ink 15,0,26 [ENTER]
```

In this case 15 is the **PEN** number, while 0 is the colour black, and 26 is the alternate colour bright white.

Finally, it is possible to make the **B O R D E R** flash between two colours by adding an extra colour number to the **B O R D E R** command. Type in:

```
border 6,9 [ENTER]
```

You will now see that the **B O R D E R** is flashing between bright red and green.

Now reset the computer [CTRL] **[SHIFT]** [ESC]

DEMONSTRATION PROGRAM

For further demonstration of the colours available, type in the following program, then run it.

We have included some sound in the program. This will be explained in a later section.

```
10 mode 0: ink 0,2:ink 1,24: paper 0 [ENTER]
20 pen 1: for b=0 to 26: border b[ENTER]
30 locate 3,12:print"BORDER COLOUR";B [ENTER]
40 sound 4,(40-b> [ENTER]
50 for t=1 to 600:next t:next b:cls [ENTER]
60 for p=0 to 15:paper p:pen 5:print "paper";
  p:print [ENTER]
70 for n=0 to 15:pen n:print "pen";n [ENTER]
80 sound 1,(n*20+p) [ENTER]
90 for t=1 to 100:next t:next n[ENTER]
100 for t=1 to 1000:next t:cls:nextp [ENTER]
110 cls:paper 0: pen 1:locate 7,12:print
    "THE END" : for t=1 to 2000: next t[ENTER]
120 mode 1: border 1:ink 0,1:ink 1,24:paper 0 :
    pen 1 [ENTER]
```

```
run [ENTER]
```

GRAPHICS

From this point on, we will not ask you to press the [ENTER] key after each line. We will just assume that you will do it automatically.

There are a number of character symbols in the computer's memory. To print any one of these, use the key word `chr$()`. Inside the brackets should be the symbol number, which is in the range from 32 to 255.

Press [CTRL][SHIFT] and [ESC] to reset the computer, then type in:

```
print chr$(250)
```

On the screen you will see character number 250, which is a man walking to the right.

To see all the characters and symbols appear on the screen with their associated number, type in the following program, remembering to press [ENTER] after each line.

```
10 for n=32 to 255: print n;chr$(n);  
20 next n  
run
```

For your reference, the range of characters together with their respective reference numbers, appear in Appendix III at the back of this book.

LOCATE

This command is used to reposition the character cursor to a specified part of the screen. Unless changed by the `Locate` command, the character cursor starts at the top left corner of the screen, which corresponds to x, y co-ordinates 1,1 (x is the horizontal position and y is the vertical position). In mode 1 there are 40 columns and 25 lines. To position a character in the centre of the top line in mode 1, we would use 20,1 as the x,y co-ordinates.

To see this, type in: (remember to [ENTER] each line)

```
mode 1           ..... ..screen clears, cursor moves to top left
```

```
10 Locate 20,1  
20 print chr$(250)  
run
```

Just to prove that this is on the top line, type in:

```
border 0
```

The BORDER will now be black and you will see the man at the middle of the top line of the screen.

In mode 0, there are only 20 columns, but the same 25 lines. If you now type in:

```
mode 0  
run
```

You will see that the man now appears at the top right corner of the screen. This happens because the x co-ordinate 20, is the last column in mode 0.

In mode 2, there are 80 columns and 25 lines. Using the same program, you will probably be able to guess where the man will appear. Type in:

```
mode 2  
run
```

return **to mode 1** by typing in:

```
mode 1
```

Now experiment for yourself, modifying the `locate` and `chr$()` numbers to position various characters anywhere on the screen. Just for example, type in:

```
locate 20,12:print chr$(240)
```

You will see an arrow in the centre of the screen. Note that in this instruction

20 was the horizontal (x) co-ordinate (in the range 1 to 40)

12 was the vertical (y) co-ordinate (in the range 1 to 25)

240 was the character symbol number (in the range 32 to 255)

To get the character symbol 250 to be repeated across the screen, type in the following program:

```
5 cls  
10 for x = 1 to 39  
20 locate x,20  
40 print chr$(250)  
50 next x  
60 goto 5  
run
```

Press **[ESC]** key twice to break

In order to remove the previous character from the screen before printing the next character, type in:

```
40 print " "; chr$(250)
```

(This new line 40 automatically replaces the line previously typed as line 40.)

Now type in:

run

To improve the movement of the character across the screen, add the following line:

```
30 call &bd19
```

This program can be further enhanced to improve the movement by adding some delay loops and by using a different returning character symbol.

Type in:

list

Now add the following lines to the program:

```
60 for n = 1 to 300 : next n
65 for x = 39 to 1 step -1
70 locate x,20
75 call &bd19
80 print chr$(251);" "
85 next x
90 for n = 1 to 300:next n
95 goto 10
run
```

Try this interesting small program. We have added some other commands that will be explained in later chapters. For now, just type in:

```
new
10 mode 1
20 locate 21,14:print chr$(244)
30 tag
40 for x=0 to 624 step 2
50 mover -16,0
60 if x<308 or x>340 then y=196:goto 90
70 if x<324 then y=x-104:goto 85
80 y=536-x
85 sound 1,0,20,7
90 move ox, oy:print " ";:ox=x:oy=y
100 move x,y
110 if (x mod 4) = 0 then print chr$(250);
    else print chr$(251);
120 for n=1 to 4: call &bd19:next n
130 next x
140 tagoff
150 goto 20
run
```

PLOT

Unlike the locate command, plot is used to determine the position of the graphics cursor, using pixel co-ordinates. (A pixel is an extremely small segment of the screen).

Note that the graphics cursor is not visible and is different to the character cursor.

There are 640 horizontal pixels by 400 vertical pixels. The **x,y** co-ordinates are positioned with respect to the bottom left corner of the screen, which has **x,y** co-ordinates of 0,0. Unlike the locate command use-d for characters, the co-ordinates do not differ in modes 0,1, or 2.

To see this, first reset the computer using **[CTRL] [SHIFT] [ESC]**, then type in (remembering to **[ENTER]**) :

```
plot 320,200
```

A small dot will appear in the centre of the screen.

Now change the mode by typing:

```
mode 0  
plot 320, 200
```

You will see the dot is still in the centre but is now larger. Change the mode again and type in the same command to see the effect in mode 2. Type in:

```
mode 2  
plot 320, 200
```

The dot is still in the centre, but it is now much smaller.

Plot several dots over the screen in various modes, in order to accustom yourself with this command. When you have finished, return to mode 1 and clear the screen by typing in:

```
mode 1
```

DRAW

First reset the computer using **[CTRL][SHIFT]** and **[ESC]**. The draw command draws a line from the current graphics cursor position. To see this in more detail, draw a rectangle on the screen by using the following program. We start by repositioning the graphics cursor with a plot command. Then drawing a line from the graphics cursor position, up towards the top left corner, then from here to the right corner etc.

Type in:

```
5 cls
10 plot 10,10
20 draw 10,390
30 draw 630,390
40 draw 630,10
50 draw 10,10
60 goto 10
run
```

Press [ESC] twice to break from this program.

Now add the following lines to the program, to draw a second rectangle inside the first. Type in:

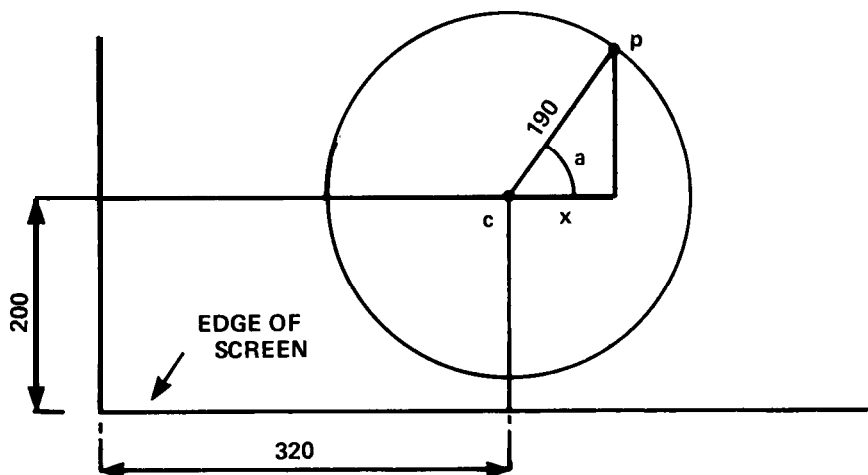
```
60 plot 20,20
70 draw 20,380
80 draw 620,380
90 draw 620,20
100 draw 20,20
200 goto 10
run
```

Press [ESC] twice to break from this program.

CIRCLES

Circles can either be plotted or drawn. One method of forming a circle is to plot the x,y, co-ordinates of each point on the circumference of a circle. Refer to the figure below and you will see that point p on the circumference can be plotted using x and y co-ordinates. These are:

$$x=190*\cos(a) \text{ and } y=190*\sin(a)$$



NEW

We have now started to use the keyword **n e w** before typing in the program itself. This tells the computer to fully clear the memory in a similar way to reset (**[CTRL]** **[SHIFT]** and **[ESC]** keys). However, it differs in that the screen is not cleared; only the memory. This is useful if you want to keep the old program on the screen for reference when writing a new program.

Plotting the points of a circle.

In the previdus program we plotted points with respect to the bottom left corner of the screen. If we now wanted to position a circle in the centre of the screen we would have to plot the centre of the circle at co-ordinates 320,200, then position all points of the circle relative to the centre position, by adding on the centre position co-ordinates.

A program to draw a circle would then be like this. Type in:

```
new  
5 cts  
10 for a=1 to 360  
15 deg  
20 plot 320, 200  
30 plot 320+190*cos(a), 200+190*sin(a)  
40 next  
run
```

The radius of the circle can be reduced by lowering the 190 figure (190 refers to pixels).

To see the effect of the circle being plotted differently (in radians), delete line 15 from the program by typing in:

15

To see a solid circle drawn by lines from the centre, edit line 30, replacing the word **p L o t** with the word **d r a w**. Line 30 will then be:

```
draw 320+190*cos(a), 200+190*sin(a)
```

Try this with and without line 15 again.

You will note that line 40 of this program is **n e x t** instead of **n e x t a**

It is permissable to simply type in **n e x t** in programs. The computer will work out which for expression the **next** is to be associated with. In programs where there are numerous **f o r** and **n e x t** loops you may wish to add the appropriate name after the word **n e x t** in order to identify the **n e x t** statement when studying the program.

ORIGIN

In the previous program we used the `plot` command to plot the centre of a circle, then added the `x,y` co-ordinates to this centre position. Instead of adding these position co-ordinates to the point plotted, we can use the `origin` command. This will position the centre of the circle, then position the `x,y` co-ordinates of all the points on the circumference (in 1 degree steps) from the origin. To see this type in:

```
new  
5 cls  
10 for a = 1 to 360  
15 deg  
20 origin 320, 200  
30 plot 190*cos(a), 190*sin(a)  
40 next  
run
```

Again, you can alter lines **15** or **30** to remove `d eg` or `d raw` the solid circle from the centre.

To plot four smaller circles on the screen, type in the following program:

```
new  
5 cls  
10 for a=1 to 360  
15 deg  
20 origin 196, 282  
30 plot 50*cos(a), 50*sin(a)  
40 origin 442, 282  
50 plot 50*cos(a), 50*sin(a)  
60 origin 196, 116  
70 plot 50*cos(a), 50*sin(a)  
80 origin 442, 116  
90 plot 50*cos(a), 50*sin(a)  
100 next  
run
```

Once again, you can remove line 15 and modify lines 30,50,70 and 90 to use **the** `DRAW` command.

GOSUB RETURN

If there are a set of instructions within a program which are to be carried out a number of times, these instructions can be typed as a sub-routine, and can be called into action by the command `gosub` followed by the line number.

The end of a `sub` routine is marked by typing in the instruction `return`. At this point, the computer will return to the line that followed the `gosub` command which it had just obeyed.

In the previous program, the instruction `plot 50*cos(a),50*sin(a)` was repeated 4 times. This instruction can be typed in as a sub-routine; and called into action each time it is needed by using the word `gosub`. **Type** in the following:

```
new
5 cls
10 for a=1 to 360
15 deg
20 origin 196,282
30 gosub 120
40 origin 442,282
50 gosub 120
60 origin 196,116
70 gosub 120
80 origin 442,116
90 gosub 120
100 next
110 end
120 plot 50*cos(a),50*sin(a)
130 return
run
```

Note that the instruction `end` is used in line 110; otherwise the program would naturally continue after instruction 100, and carry out instruction 120, which is only required when called by `gosub`.

To conclude this section, try the following program which incorporates a lot of the programming commands and keywords that you should now understand. **Type** in:

```
new
10 mode 0:border 6:paper 0:ink 0,0
20 gosub 160:for x=1 to 19:locate x,3
30 pen 15:print" ";chr$(238)
40 for t=1 to 50:next t:sound 2,(x+100)
50 next x:gosub 160:for b=3 to 22
60 locate 20,b:pen 7:print chr$(252)
70 cls:gosub 160:next b
80 sound 2,0,100,15,0,0,1
90 gosub 160:border 16,24:locate 20,25
100 pen 14:print chr$(253);
110 for t = 1 to 1000:next t
120 border 6:gosub 160:for f=3 to 24
130 locate 10,(25-f):pen 2
140 print chr$(144):cls:gosub 160
150 sound 7,(100-f),5:next f:goto 10
160 locate 10,25:pen 12
170 print chr$(239):return
run
```

SOUND

Sound effects are generated by a loudspeaker within the computer itself. If you are using the **MP1** modulator power supply and a domestic television, turn the TV's volume control to a minimum.

The level of sound can be adjusted by use of the **VOLUME** control on the right hand end of the computer. The sound can also be fed to the auxiliary input socket of your stereo system, using the (I/O) socket at the left hand end of the computer back panel. This will enable you to listen to the sound generated by the computer in stereo, through your **hi-fi** loudspeakers or headphones.

The **SOUND** command has seven parameters. The first two of these must be used, the rest are optional. The command is typed in as:

SOUND channel status, tone period, duration, volume, volume envelope, tone envelope, noise period.

In the following examples, we will type in 1 as the channel status - in other words, its reference number.

tone PERIOD

Refer to Appendix VII and you will see that the note middle c, has a tone period of 478. Type in:

```
new
10 sound 1,478
run
```

You will hear a short note which is middle c lasting 0.2 second.

DURATION

When no duration of the **sound** is specified it will last 0.2 second. The unit of duration is in 0.01 second. To make a note last 1 second, 100 would be used; to last 2 seconds, 200 would be used. Type in:

```
10 sound 1,478,200
run
```

You will hear the note middle c lasting 2 seconds.

VOLUME

This number specifies the starting volume of a note. The number is in the range 0 to 7. If however, a volume envelope is specified, the range is extended from 0 to 15. If no number is used, 4 is assumed. Type in:

```
10 sound 1,478,200,3
run
```

Note the volume of this **sound** . **Now** type it in using a higher volume number:

```
10 sound 1,478,200,7
run
```

You will now hear that this is much louder.

VOLUME ENVELOPE

The volume envelope command is **e n v**. This normally has 4 parameters: The last 3 parameters may appear in any of up to 5 optional envelope sections available. We are only using one of these here. Further explanation will appear in chapter 6.

e n v envelope number, number of steps, amplitude (size) of step, step time.

ENVELOPE NUMBER

This is the number given to a particular envelope so that it can be specified in the sound command. The range of envelope numbers is 0 to 15.

NUMBER OF STEPS

This is used in conjunction with the step time. For example, you may wish to have 10 steps of 1 second each. In such a case the number of steps is 10. The range of step numbers is 0 to 127.

AMPLITUDE OF STEP

Each step can vary in amplitude from a level of 0 to 15 with respect to the last step. The 15 volume levels are the same as those in the sound command. However, the step can be adjusted from -128 to +127 so that you can not only vary the amplitude up or down in the obvious way, but can vary it by using numbers higher than 15 to give some strange effects. The range of amplitude of step numbers is -128 to + 127

STEP TIME

This number specifies the time between steps in 0.01 second, (1/100th's of a second) units. The range of step time numbers is 0 to 256. The longest time between steps is therefore 2.56 seconds.

To experiment with the volume envelope, type in the following program:

```
5env 1,10,1,100  
10 sound 1,284,1000,1,1  
run
```

Line 10 specifies a sound with a tone period of 284 (international a), lasting for 10 seconds with a start volume of 1, and using volume envelope number 1, shown in line 5 consisting of 10 steps, raising the volume of each step by 1, every 1 second (100 x 0.01 second).

Change line 5 in each of the following ways and then **run** each time to hear the effect of changing the envelope.

```

5 env 1,100,1,10
5 env 1,100,2,10
5 env 1,100,4,10
5 env 1,50,20,20
5 env 1,50,2,20
5 env 1,50,15,30

```

And finally try this:

```

5 env 1,50,2,10

```

You will notice that half way through the sound, the level remains constant. This is because the number of steps was 50 and the time between each step was 0.1 second. Therefore the length of time during which the amplitude varied was only 5 seconds, but the duration of the sound in the `s o u n d` command in line 10 was 10 seconds (number 1000).

Try experimenting yourself, to see what type of sounds you can create.

TONE ENVELOPE

The tone envelope command is **e n t**.

This normally has 4 parameters.

The last 3 parameters may appear in any of up to 5 optional envelope sections available. We are only using one of these here. Further explanation will appear in Chapter 6.

e n t envelope number, number of steps, tone period of step, step time.

ENVELOPE NUMBER

This is the number given to a particular envelope so that it can be specified in the `s o u n d` command. The range of envelope numbers is 1 to 15.

NUMBER OF STEPS

This is used in conjunction with the step time. For example, you may wish to have 10 steps of 1 second each. The range of step numbers is 0 to 239.

TONE PERIOD OF STEP

The tone period for each step can vary between -128 to +127. Negative steps increase the frequency of the notes (make the notes higher). The shortest tone period is 0. This must be remembered when calculating the tone envelope. The full range of tone periods is shown in Appendix VII. The range of tone period of step numbers is -128 to + 127.

STEP TIME

This number specifies the time between steps in 0.01 second (1/100th's of a second). The range of a step time number is 0 to 255. The longest time between steps therefore, is 2.55 seconds.

To experiment with the tone envelope, type in the following program:

```
5 ent 1,100,2,2
10 sound 1,284,200,7,0,1
run
```

Line 10 specifies a sound with a tone period of 284 (international a) lasting for 2 seconds with a start volume of 7, without a volume envelope (represented by 0), and with tone envelope number 1.

Line 5 is tone envelope number 1 consisting of 100 steps, increasing the tone period (reducing the frequency) by 2 every 0.02 second (2/100th's of a second).

Now change line 5 in each of the following ways and then run each time to hear the effect of changing the tone envelope:

```
5 ent 1,100,-2,2
5 ent 1,10,4,20
5 ent 1,10,-4,20
```

Now replace the `sound` command and the tone envelope by typing in:

```
5 ent 1,2,17,70
10 sound 1,142,140,15,0,1
15 goto 5
run
```

Press the [ESC] key twice to break

Now you can put the volume envelope, tone envelope, and `sound` command together to create various sounds. Start by typing in:

```
new
5 env 1,100,1,3
10 ent 1,100,5,3
20 sound 1,284,300,1,1,1
run
```

Then replace line 10 by typing in:

```
10 ent 1,100,-2,3
```

Now replace all the lines by typing in:

```
5  env 1,100,2,2
10 ent 1,100,-2,2
20 sound 1,284,200,1,1,1
run
```

Try some more variations for yourself.

NOISE

Noise can be added to the end of the **sound** command. A range of noise is available in the range 1 to 31. Try this by adding the noise number at the end of the **sound** command still using the **env** command.

Replace lines 5 and 20 by typing in:

```
5  env 1,100,3,1
20 sound 1,200,100,1,1,1,5
run
```

Again, try to get some unusual sounds by modifying the volume envelope and the **sound** command, with and without noise.

