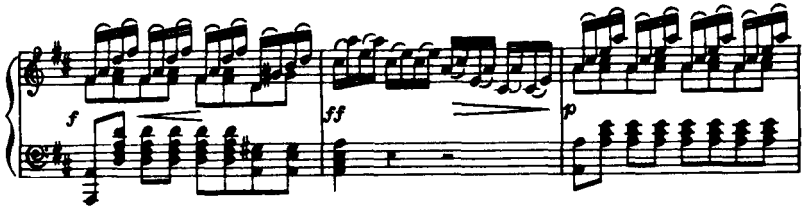
6 Sound primer

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 *CPC464* sounds as good as it looks. To get the most from the imaginative SOUND processing software, you need to understand ***the*** philosophy behind the timing structures.



Subjects covered in this chapter:

* Tone periods
* Sound command
* Enveloping
* Queuing and **synchronisation**

If all you want to do is to make the computer 'bleep' - then type

```
PRINT CHR$(7)
```

And go no further....but you will be missing some of the most interesting and rewarding features of the CPC464, so this section is a primer that ranges across an overall view of the subject followed by a detailed look at the keywords and how to specify them. It will furnish the programmer with the basics on how to build a scale of notes, invent various types of musical instruments, and structure tunes using them.

6.1 Fundamentals of SOUND

When you hear the sound created by a standard note of a tune there are several characteristics to be considered:

1) Pitch, and variations of pitch during the time of the note.
2) Volume, and variations of volume during the time of the note.
3) Length of the note.

6.2 Pitch

In a musical note the pitch is by far the single most important factor. A musical note can be described as a regular oscillation, all oscillations have frequency, period and amplitude. (see fig 1).

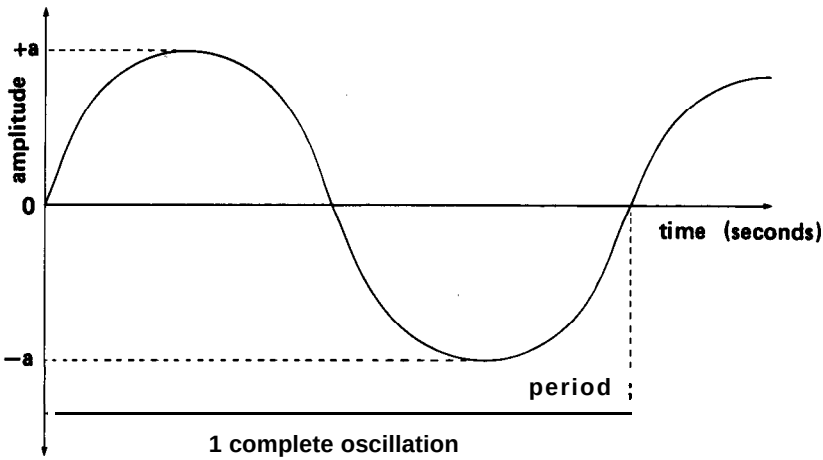


Figure 1: The characteristics of a musical note

The frequency is the number of oscillations per second, and the period is length of time each oscillation lasts. Amplitude is an attribute of volume and is not relevant here in defining the pitch of a note.

Frequency and period are related to each other by the simple formula:

Frequency (expressed in Hertz) = l/Period (expressed in seconds)

The relationship between frequency and the ‹tone period) in the S 0 **U N D** command is:

Frequency (expressed in *Hz)* ▫ 125000 / tone period)

Thus a tone period) of 1000 would result in a 125 Hz tone, and a tone period) of 125 would result in a **1000Hz (1kHz)** note.

Either of these could be used to set the pitch, but Amstrad BASIC chooses to use the tone period,. Don't be mislead by the fact that as the period goes up in value, the pitch goes down. The pitch appears in the keyword description of S 0 U N D as tone period). The range of tone periods available is extensive (0....4095) and must be expressed as an integer. This may lead to some rounding errors when constructing a scale of musical notes, but nothing that would offend any but the most critical ear -see Appendix VII.

When a note is played on a real musical instrument the pitch may vary, sometimes deliberately by vibrato. Using the E NT command (**E** Nvelope Tone) we can design a specific format for the structure of period changes during each note - and this may be invoked in the **SOUND** command. This feature is known here as the tone envelope,.

6.3 Volume

Volume is simply the measure of how loud the the sound is. To set the initial volume level of each S 0 U N D on the CPC464, there is a straightfoward scale :
increase in value = increase in volume level, set by an integer number
in the range 0....15 (0...7 if no volume envelope is specified).

This value is also established in the SOUND command and is refered to as volume,.

If you have already tried some simple notes, you will have noticed the relatively unexciting nature of the way in which a regular note with no 'processing' is reproduced. This is because in conventional musical instruments, as each note is played there is an increase in volume at the start of each note - called the attack -and a decrease in volume at the end of the note - called the decay.

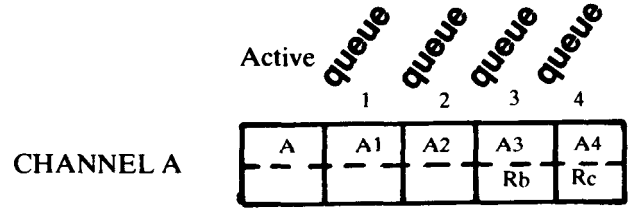
The different forms of attack and decay for each type of musical instrument can be simulated on the computer by adjusting the volume characteristics within a volume envelope,

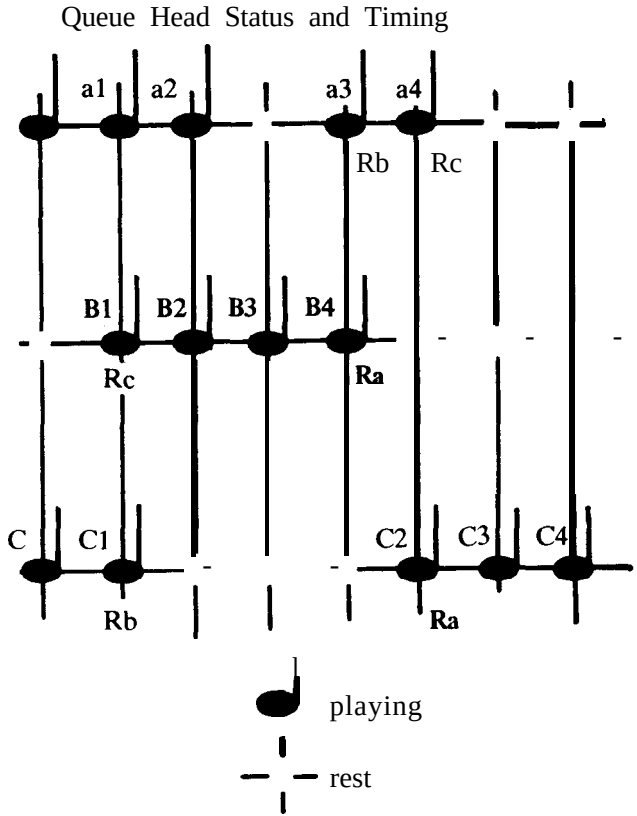
6.4 Length of the note

The length of note is a basic feature of any musical construction, the unit of musical note length is the crotchet, and there are minims, semiquavers etc. that are simple multiples or fractions of the basic unit. However tunes can be played at varing speeds - and the basic length of time or reference period for the 'standard' crotchet has to be determined.

To do this there is an expression in the SOUND command known as ‹duration›, which is an integer where 1 unit = 1/100th second (i.e. 100 = 1 sec.). Note that it may also also be determined by the length of the volume envelope), which helps to avoid confusion when a volume envelope, is used.

Rendezvous for sound channels





A-A4
B-B4
c-C4
} Sound commands

Ra – rendezvous with channel A
Rb – rendezvous with channel B
Rc – rendezvous with channel C

6.5 Other SOUNDs

Random noise ('white noise') is a S 0 UN D and can be generated on the computer, but it is not a musical note. It can be added to the background of tunes to create interesting variations, or simply used on its own for special effects - and explosion is basically white noise with volume envelope, control. This noise varies the frequency randomly around a period that is set in the S 0 U N D command parameter noise period). An important feature is that although there are three separate channels available with the ability to set three tone periods), only one noise period, can be set to appear on the required combination of channels.

6.6 Multiple SOUND and channels

Most pieces of music are written with at least two 'clefs', bass and treble. To enable this approach on the CPC464 there have been three S 0 UN D channels provided -A,B and C. These can all play independently, or be timed to coincide when required. Selection of channels is made in the S OU N D command parameter channel status.

6.7 Channel queues

Each S 0 U N D channel has a queue of S 0 U N Ds to play. There is space in this queue for five separate SO U N D commands: one active and four waiting. The operating system of the CPC464 can continue with other tasks while playing out the sound queue, only returning when necessary to pick up more S 0 U N D commands.

6.8 Channel status

The keyword SQ is used to determine the state of the channel that you wish to interrogate, and it returns information about free places in the queue, 'rendezvous' and holds. 0 N S Q G 0 S U B is an interrupt command used to bring the attention of the computer back to the S 0 U N D generation section of the program.

6.9 Rendezvous and holds

To force channels to synchronize there is a 'RENDEZVOUS facility. This is where a marker is set on two or more channels that force the simultaneous action of those SOUND commands. These are very useful to reset the timing of continious notes where a tune contains breaks or 'rests', not occurring simultaneously on both channels.

HOLD (see the S 0 U N D channel status,) is important where an overall synchronization of channels is required (i.e. at the start of tune), maybe with a delay, and it is necessary to have the queues primed ready. Then a HOLD and RELEASE sequence is used.

6.10 Sound generation command sequence

Form of the command: S 0 U N D G,H,I,J,K,L,M

Where....
G: Channel Status
H: Tone Period
I: Duration
J: Volume
K: Volume Envelope
L: Tone Envelope
M: Noise Period

SOUND is a command where the parameters G to M are all integers and only the first two are mandatory. The remaining parameters are optional, but have default values that will be discussed under the parameter descriptions.

6.10.1 Parameter descriptions:

G: Channel Status

Value range 1.....255
default if omitted: none (mandatory entry>

In the CPC464 it is possible to play up to three different S 0 U N Ds at once. This is achieved by having three S 0 U N D channels (or queues), referred to as A, B and C. The input integer range is given in decimal form, but when converted to an 8 bit value (thus yielding a bit significant result), the active bits specify the following commands:

DECIMAL	BIT	COMMAND
1	0 LSB	send SOUND to channel A
2	1	send SOUND to channel B
4	2	send SOUND to channel C
8	3	rendezvous with channel A
16	4	rendezvous with channel B
32	5	rendezvous with channel C
64	6	hold
128	7 MSB	flush

(*LSB* and *MSB* signify Least Significant Bit, and Most Significant Bit in the table above.)

integer entry examples:

2 = send the following S 0 UN D to output channel B
5 = send the following S 0 UN D to output channels A and C
98 = 64 + 32 + 2
= send the following SOUND to output channel B, rendezvous with channel C and hold.

It is important to remember that if a rendezvous is to be arranged between two (or more) channels, then it is necessary to flag on the channel status) of each channel at the appropriate time.

The hold on any combination of channels, stops the processing of that command and freezes the queue behind it, until it is freed by a R E LEA S E command (or 'flushed by a later S 0 U N D command).

When the flush bit is activated with a channel(s) the S 0 UN D put through with the flush is excecuted immediately leaving the queue empty and the head of the channel inactive. Any S 0 U N D s currently being played are terminated.

H: Tone Period

Value range 0....4095

default if omitted : none (mandatory entry)

Enters a period that sets the frequency of the S 0 UN D to be played (pitch of the note). The frequency can be evaluated from the formula :

frequency = 125000 / period

By using 0, no frequency is set, this is most useful where only a 'noise' is required.

I: Duration

Any integer value in the permissable range -32768....+32767

default if omitted : 20

For values greater than zero the value entered represents the duration in 1/100ths second. When equal to zero, the duration is governed by the length of the volume envelope specified.

Where the duration is less than zero, the positive value of this number gives the number of times the volume envelope specified should be repeated.

J: Volume

An integer value in the range O.... 15 (0.. ..7 where no volume envelope is specified)

default if omitted : 12 (4 where no volume envelope is specified)

This is the starting volume, it can be altered by the volume envelope if one is specified. Zero is no volume.

K: Volume Envelope

An integer value in the range 0....15

default if omitted : 0

The value used specifies a pre-defined envelope. To define an envelope use the E NV command. A permanent definition is volume <envelope number) 0, this cannot be changed by the E NV command and is set to 2 seconds constantly at the 'volume level'.

L: Tone Envelope

An integer value in the range 0.. ..15

default if omitted : 0

The value used specifies a pre-defined envelope. To define an envelope use the E N T command, A permanent definition is tone <envelope number 0, this cannot be changed by the E NT command and is set to the 'tone period' constantly.

M: Noise Period

An integer value in the range 0.. ..15

default if omitted : 0

Specifies the noise to add to the S 0 U N D. If default or zero is used, then no noise is added. Remember that only one noise period can be set at one time. This means that all channels specified for noise will receive the same noise.

6.11 The E NV command and volume envelopes

Attack and decay are required to add dimension to the note and give it life. This command (E Nvelope Volume) enables the shape of a note to be formed. It is a good idea before attempting to describe the shape to the computer using the numeric instruction sequence to sketch out your requirement on paper. See example below :

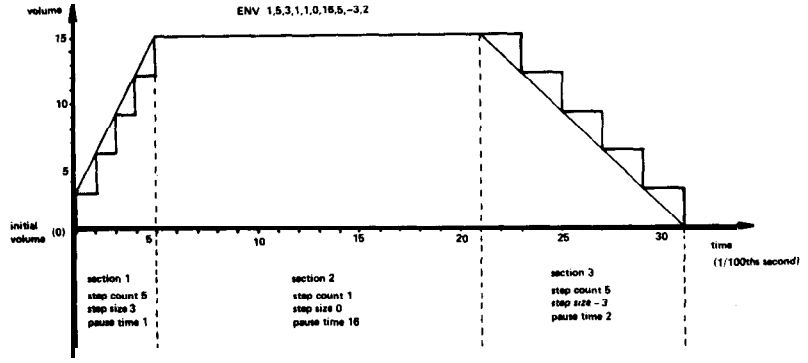


Figure 3: Volume envelope example

The shape must be plotted in numbers so that it can be formatted into a command statement. To do this , subdivide the shape into sections , you may have up to 5. But in each section the shape must be a straight line. Now divide each section into steps, the number of steps you choose will be known as the ‹**step count**›, these steps will be given a length of time called ‹**pause** time,. 1 = 1/100th sec.].

The steps will also have an increase or decrease in volume known as the step **size**›. If no steps are specified in a section then there is just a volume setting carried foward to the next section.

N.B. When attempting to simulate a musical instrument it is often the case that the initial and final volume of each note will be zero, so the initial volume setting in the S 0 U N D command should be zero and all the volume be added by the envelope.

Form of the command:

ENV N,P1,Q1,R1,P2,Q2,R2,P3,Q3,R3,P4,Q4,R4,P5,Q5,R5

N: Envelope Number value range 1....15
P1....5 : Step Count (sections l.... 5) value range 0 127
Q1.... 5 : Step Size (sections l.... 5) value range -128....+127
R1....5 : Pause Time (sections 1.... 5) value range 0 255

The (envelope number, is a compulsory label. A complete section is mandatory, but the number of sections is optional. For example, omitting sections 4 and 5 assumes only 3 sections.

6.12 The ENT command and tone envelopes

These have exactly the same type of construction as volume envelopes, only this feature creates small variations in frequency of a note, in other words, a form of vibrato. After deciding on the shapes of the note's frequency, sketch it as before and divide into sections and steps. see example below :

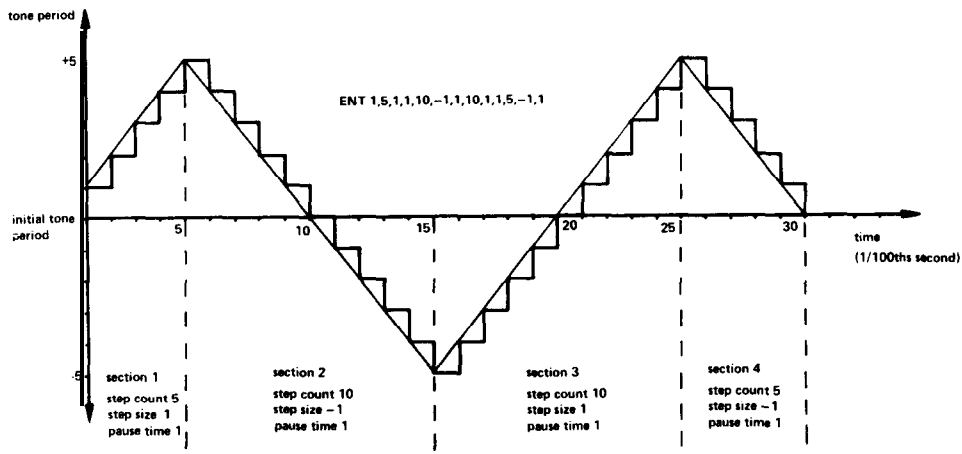


Figure 4: Characteristics of a tone envelope

The major difference between volume envelopes and tone envelopes is that tone envelopes have no effect on the duration of a note previously defined - either in the S 0 U N D command or in a volume envelope. If the tone envelope finishes before the note then the final tone period continues. However, if it overruns the time scale of the note then the remaining steps' sections are abandoned.

To repeat a tone envelope for the duration of the sound, use a negative envelope number. (Note It is not called up as negative in the S 0 UN D command).

Form of the command:

ENT S,T1,V1,W1,T2,V2,W2,T3,V3,W3,T4,V4,W4,T5,V5,W5

S: Envelope Number value range 1....15 (-ve for repeat)
T1.... 5 : Step Count (sec. 1....5) value range 0....239
V1.... 5 : Step Size (sec. 1....5) value range -128....+127
W1....5: Pause Time (sec. 1....5) value range 0....255 (1/100ths second)

S is mandatory, and completion of each individual section is also mandatory.

When an <envelope number is defined, all previous settings are overruled. Specifying an envelope of either type without any sections will reset the settings of that <envelope number, to zero.

6.13 Other associated functions and commands

SQ(x)

X is the channel number :1,2 or 4

These values represent the channels A, B, and C respectively. As shown in the 'bit significant' table of channel status, a parameter of the SOUND command. This is an inquiry into the state of the SOUND channel nominated.

It is a function returning an answer which is an integer in the range 0.....255. To extract the information from this number the following bits are *significant* for the data:

Bit 0....2 number of free spaces in the queue tested (value 0....4)
Bit 3 rendezvous with channel A showing at head
Bit 4 rendezvous with channel B showing at head
Bit 5 rendezvous with channel C showing at head
Bit 6 hold at head of queue
Bit 7 channel currently playing

Bits 3 to 6 (rendezvous and hold) were described in the channel status section. The only other facility of this test is to disable the ON S Q GO S UB command described in the next paragraph.

ON SQ GOSUB

ON SQ(y) GOSUB ‹line number›

y is the channel number :1,2 or 4

This is an interrupt command that is armed and detects when there is a gap in the S 0 UN D queue nominated, then fires into the SOUND subroutine. The S Q and SOUND keywords both disarm the interrupt. The subroutine ends with a R ET U RN in the normal way. All channels have the same priority and when there is a space in the nominated channel the command will 'fire', thus disarming itself at the same time, Hence the subroutine will need to reset the interrupt, if it is required again.

RELEASE

RELEASE z

z indicates the channel number(s), an integer value in the range 1.....7, which is bit significant.

As described in the S 0 UN D command parameter channel status) it is possible to flag a hold on a channel with a particular SOUND command. The R E LEAS E command is simply to remove these holds. The channel input is bit significant, and when decoded gives the combination of channels to be released. Release of a channel not held has no effect.

Bit 0 : Channel A
Bit 1 : Channel B
Bit 2 : Channel C