

Chapter 14 - SN76489

The SN76489 is a compact square-wave and noise chip on the IE audio bus. It has three tone channels, one noise channel, and one byte-stream command port. It is useful for arcade beeps, bass pulses, arpeggios, and sharp noise percussion.

The chip is not programmed through a BASIC keyword. Use POKE8 to send command bytes to the port.

```
10 REM SN FIRST TONE
20 POKE32 &H00F0800,1
30 REM LATCH CHANNEL 0 DIVIDER
40 D=254
50 POKE8 &H00F0C30,&H80+(D AND 15)
60 POKE8 &H00F0C30,INT(D/16) AND 63
70 REM ATTENUATION 0 IS LOUDEST
80 POKE8 &H00F0C30,&H90
```

Expected result: channel 0 plays a bright square wave. D is the 10-bit tone divider. &H90 sets channel 0 attenuation to 0, the loudest setting.

Lines 50 and 60 write one divider through two command bytes. The latch byte contains the channel number and the low four divider bits; the following data byte supplies the high six bits. Line 80 is a second latch byte, this time for channel 0 attenuation.

Try changing D to 508. The tone drops by one octave.

14.1 Shape of the chip

Item	Value
Tone channels	3, numbered 0 to 2
Noise channel	1, numbered 3
Tone divider	10 bits per tone channel
Volume format	4-bit attenuation, 0 loudest, 15 silent
Noise forms	Periodic and white
Access method	One byte-stream write port
Ready status	Bit 0 of SN_PORT_READY

All channels power up silent because every attenuation register starts at 15.

14.2 Register block

The SN76489 block occupies \$F0C30 to \$F0C3F. Only the first three addresses have defined behaviour.

Address	Name	Purpose
\$F0C30	SN_PORT_WRITE	Write one SN76489 command byte

Address	Name	Purpose
\$F0C31	SN_PORT_READY	Read bit 0; 1 means ready
\$F0C32	SN_PORT_MODE	0 selects 15-bit noise, 1 selects 16-bit noise

Use POKE8 for command bytes. A 32-bit write to SN_PORT_WRITE uses only its low byte.

14.3 Command bytes

Every byte written to SN_PORT_WRITE is either a latch byte or a data byte.

Bit field	Latch byte meaning
Bit 7	1, marks this as a latch byte
Bits 6 to 5	Channel number, 0 to 3
Bit 4	0 tone/noise control, 1 attenuation
Bits 3 to 0	Low four data bits

If bit 7 is clear, the byte is a data byte. For tone channels it supplies the upper six bits of the divider for the most recently latched tone channel. For attenuation it supplies a new four-bit attenuation value for the most recently latched channel. Data bytes after a noise-control latch are ignored.

The base latch bytes are:

Channel	Tone or noise latch	Attenuation latch
0	&H80 + low4	&H90 + atten
1	&HA0 + low4	&HB0 + atten
2	&HC0 + low4	&HD0 + atten
3	&HE0 + noise	&HF0 + atten

14.4 Tone dividers

A tone divider is 10 bits:

```
divider = low4 + 16 * high6
frequency = clock / (32 * divider)
```

Divider zero is treated as divider 1024, matching the long-period silence and sub-bass behaviour of the chip family.

Useful clock values are:

Clock style	Value in Hz
NTSC	3579545
PAL	3546893
BBC	4000000

This listing plays a small arpeggio by changing channel 0's divider:

```

10 REM SN ARPEGGIO
20 POKE32 &H000F0800,1
30 REM TURN CHANNEL 0 ON
40 POKE8 &H000F0C30,&H90
50 FOR I=0 TO 95
60 REM CHOOSE ONE OF FOUR DIVIDERS
70 N=I-INT(I/4)*4
80 IF N=0 THEN D=254
90 IF N=1 THEN D=201
100 IF N=2 THEN D=169
110 IF N=3 THEN D=127
120 POKE8 &H000F0C30,&H80+(D AND 15)
130 POKE8 &H000F0C30,INT(D/16) AND 63
140 FOR Q=1 TO 40
150 NEXT Q
160 NEXT I
170 POKE8 &H000F0C30,&H9F

```

Expected result: the tone steps through a repeating four-note pattern, then line 170 silences channel 0.

Line 40 makes channel 0 audible before the loop starts. Lines 70 to 110 turn the loop counter into a repeating index from 0 to 3, then choose a divider. Lines 120 and 130 send the two divider bytes. The short FOR Q loop gives each note time to be heard, and the final attenuation byte silences the channel.

Try replacing 127 with 142 in line 110. The last note becomes lower, so the pattern feels less resolved.

14.5 Noise channel

The noise latch byte is $\&HE0 + \text{value}$. The low two bits select the noise rate. Bit 2 selects white noise when set and periodic noise when clear.

Value bits	Meaning
Bits 1 to 0 = 0	Noise clock $\text{clock} / 512$
Bits 1 to 0 = 1	Noise clock $\text{clock} / 1024$
Bits 1 to 0 = 2	Noise clock $\text{clock} / 2048$
Bits 1 to 0 = 3	Use tone channel 2 divider
Bit 2 = 0	Periodic noise
Bit 2 = 1	White noise

SN_PORT_MODE selects the feedback-register length. Write 0 for the 15-bit form and 1 for the 16-bit form. Changing the mode or the noise latch resets the noise shift register.

```

10 REM SN NOISE HIT
20 POKE32 &H000F0800,1
30 REM 15 BIT WHITE NOISE
40 POKE8 &H000F0C32,0
50 POKE8 &H000F0C30,&HE4
60 REM FADE BY RAISING ATTENUATION
70 FOR A=0 TO 15
80 POKE8 &H000F0C30,&HF0+A
90 FOR Q=1 TO 40
100 NEXT Q
110 NEXT A

```

Expected result: channel 3 makes a short white-noise hit that fades to silence as the attenuation rises from 0 to 15.

Line 40 selects the 15-bit feedback register. Line 50 sends a noise latch: &HE0 selects the noise channel, and bit 2 makes it white noise. The loop then sends attenuation latches from loud to silent.

Try changing &HE4 to &HE0. The hit becomes periodic noise rather than white noise.

To lock noise to tone channel 2, set channel 2's divider first and use noise rate 3:

```

10 REM SN TONE-LOCKED NOISE
20 POKE32 &H000F0800,1
30 REM CHANNEL 2 SUPPLIES THE CLOCK
40 D=96
50 POKE8 &H000F0C30,&HC0+(D AND 15)
60 POKE8 &H000F0C30,INT(D/16) AND 63
70 REM NOISE RATE 3 FOLLOWS CHANNEL 2
80 POKE8 &H000F0C30,&HE7
90 POKE8 &H000F0C30,&HF2
100 FOR T=1 TO 3000
110 NEXT T
120 POKE8 &H000F0C30,&HFF

```

Line 80 selects white noise clocked from tone channel 2. Line 90 sets the noise channel attenuation to 2. Channel 2 does not have to be audible; its divider can act as a private clock source for the noise generator.

Try changing D in line 40 to 48. The locked noise becomes sharper because the clock source is twice as fast.

14.6 Status and side effects

PEEK8(&H000F0C31) AND 1 reads the ready bit. It is normally 1, so BASIC programs can send the next command byte immediately.

SN_PORT_MODE reads back the current noise mode. Writes to undefined addresses inside the SN block are ignored. The command port stores the most recent latch state internally, so a data byte always applies to the most recently latched tone or attenuation register.

The SN76489 has no error register. Invalid command combinations are not reported; they either update the selected latch or are ignored as described in this chapter.

14.7 Relationship to SoundChip noise

The flexible SoundChip can also generate SN-style noise on its own channels. Use this when you want SN noise mixed with envelopes, sweep, ring modulation, or sync from Chapter 12:

NOISEMODE	Algorithm
5	SN76489 15-bit white noise
6	SN76489 15-bit periodic noise
7	SN76489 16-bit white noise
8	SN76489 16-bit periodic noise

The SN76489 block remains the exact byte-stream interface. The SoundChip noise modes are a convenience path for hybrid sounds.

The next chapter covers the SID family, which adds pulse width, envelopes, and filtering.