

Chapter 12 - SoundChip and SFX

The SoundChip is Intuition Engine's IE-native synthesiser. It has ten flexible channels. Each channel has an oscillator, envelope, sweep, ring modulation, hard sync, PWM, noise selection, and DAC input. The same audio block also owns 32 SFX channels for raw sample playback.

Use the BASIC SOUND, ENVELOPE, and GATE commands for channels 0 to 3. Use POKE32 when you need the full ten-channel register map or the SFX sample trigger block.

12.1 Setup

Always enable the mixer first:

```
10 POKE32 &H000F0800,1
```

Then write channel parameters, open the gate, and let the mixer run. This first program plays a square-wave A4:

```
10 REM SOUNDCHIP FIRST NOTE
20 POKE32 &H000F0800,1
30 ENVELOPE 0,50,100,200,100
40 SOUND 0,440,200,0,128
50 GATE 0, ON
```

Expected result: channel 0 plays a sustained 440 Hz square wave. SOUND writes frequency, volume, waveform, and duty. ENVELOPE writes ADSR. GATE starts the envelope.

12.2 Channel layout

Each SoundChip channel is a \$40 byte block.

Offset	Name	Purpose
\$00	FREQ	Frequency in 16.8 fixed-point Hz.
\$04	VOL	Volume, 0 to 255.
\$08	CTRL	Bit 0 channel enable, bit 1 gate.
\$0C	DUTY	Square-wave duty, low byte.
\$10	SWEEP	Bit 7 enable, bit 3 direction, bits 4 to 6 period, bits 0 to 2 shift.
\$14	ATK	Envelope attack in milliseconds.
\$18	DEC	Envelope decay in milliseconds.
\$1C	SUS	Sustain level, 0 to 255.
\$20	REL	Envelope release in milliseconds.
\$24	WAVE_TYPE	Oscillator waveform.
\$28	PWM_CTRL	Bit 7 PWM enable, bits 0 to 6 PWM rate.

Offset	Name	Purpose
\$2C	NOISEMODE	Noise algorithm.
\$30	PHASE	Write any value to reset phase.
\$34	RINGMOD	Bit 7 enable, low nibble source channel.
\$38	SYNC	Bit 7 enable, low nibble source channel.
\$3C	DAC	Signed 8-bit DAC sample.

Channel blocks:

Channels	Base	End
0 to 3	\$F0A80	\$F0B7F
4 to 6	\$F0C40	\$F0CFF
7 to 9	\$F0D40	\$F0DFF

For channel n, the address is:

```
address = channel_base + n * $40 + offset
```

For the second and third blocks, subtract the block's first channel number before multiplying by \$40.

12.3 Waveforms and noise

WAVE_TYPE values:

Value	Waveform
0	Square, with DUTY controlling pulse width.
1	Triangle.
2	Sine.
3	Noise, with algorithm selected by NOISEMODE.
4	Sawtooth.

NOISEMODE values:

Value	Algorithm
0	White noise.
1	Periodic noise.
2	Metallic noise.
3	PSG-style noise.
4	TED 8-bit noise.
5	SN76489 15-bit white noise.
6	SN76489 15-bit periodic noise.
7	SN76489 16-bit white noise.

Value	Algorithm
8	SN76489 16-bit periodic noise.

This program plays a noise burst on channel 2:

```
10 REM SOUNDCHIP NOISE BURST
20 POKE32 &H000F0800,1
30 SOUND 2,880,180,3
40 SOUND NOISE 2,2
50 ENVELOPE 2,1,40,0,40
60 GATE 2, ON
```

Expected result: a short metallic noise burst. SOUND NOISE 2,2 writes channel 2's NOISEMODE register.

12.4 Envelope shapes

Every channel has ADSR registers. The shared ENV_SHAPE register at \$F0804 selects a default shape. Per-channel shape registers begin at \$F0860, with one 32-bit register per channel.

Value	Shape
0	Standard ADSR.
1	Saw-up rise and hold.
2	Saw-down fall and hold.
3	Looping ADSR.
4	SID-style exponential ADSR.

ENVELOPE ch,atk,dec,sus,rel writes the ADSR registers for channels 0 to 3. Use POKE32 for higher channels.

12.5 Sweep, ring modulation, and sync

SOUND SWEEP ch,enable,period,shift writes the SWEEP register for channels 0 to 3:

```
10 REM SOUNDCHIP SWEEP
20 POKE32 &H000F0800,1
30 SOUND 0,220,200,4
40 SOUND SWEEP 0,1,7,3
50 GATE 0, ON
```

The BASIC form sets the enable bit, period, and shift. Direction is available through direct register writes. To sweep down, add bit 3:

```
10 POKE32 &H000F0A90,&HF3 OR 8
```

Ring modulation multiplies one channel by another. Hard sync resets one oscillator when the source wraps.

```

10 REM RING AND SYNC
20 POKE32 &H000F0800,1
30 SOUND 0,220,180,1
40 SOUND 1,440,180,0,80
50 SOUND RINGMOD 1,0
60 SOUND SYNC 1,0
70 GATE 0, ON
80 GATE 1, ON

```

Expected result: channel 1 takes a sharper, animated tone because channel 0 modulates and syncs it.

The BASIC SOUND RINGMOD and SOUND SYNC forms write the source registers at \$F0A10 and \$F0A00. The per-channel flexible offsets \$34 and \$38 are also valid for direct POKE32; set bit 7 and put the source channel in the low nibble.

12.6 DAC mode

Writing DAC puts a signed 8-bit sample value straight onto a channel. The low byte is interpreted as signed: 0 is silence, 127 is near full positive, and 128 is full negative.

```

10 REM MANUAL DAC CLICK
20 POKE32 &H000F0800,1
30 POKE32 &H000F0A84,220
40 POKE32 &H000F0A88,1
50 POKE32 &H000F0ABC,127
60 POKE32 &H000F0ABC,128
70 POKE32 &H000F0ABC,0

```

Line 30 sets channel 0 volume. Line 40 enables the channel. Lines 50 to 70 write the DAC value.

12.7 SFX channels

The SFX block plays short raw samples from memory. The extended window has 32 channels at \$F2600 to \$F29FF, with stride \$20. The older \$F0E80 to \$F0EFF window remains as legacy aliases for channels 0 to 3, so old channel-0 examples still work.

Offset	Name	Purpose
\$00	SFX_PTR	Sample address.
\$04	SFX_LEN	Sample length in bytes.
\$08	SFX_LOOP_PTR	Loop start address.
\$0C	SFX_LOOP_LEN	Loop length.
\$10	SFX_FREQ	Playback rate in Hz.
\$14	SFX_VOL	Volume field. Values 0 to 255 set the audible level; larger values are clipped to 255.
\$16	SFX_PAN_RESERVED	Reserved pan field.
\$18	SFX_FORMAT	0 signed 8-bit, 1 unsigned 8-bit, 2 signed 16-bit.
\$1C	SFX_CTRL	Bit 0 trigger, bit 1 stop, bit 2 loop.

Reading SFX_CTRL returns status bits: bit 0 playing, bit 1 error.

This listing builds a tiny unsigned 8-bit waveform in memory and triggers SFX channel 0:

```
10 REM SFX MEMORY SAMPLE
20 POKE32 &H000F0800,1
30 BASE=&H00600000
40 FOR I=0 TO 63
50 V=80
60 IF (I AND 8)=0 THEN V=200
70 POKE8 BASE+I,V
80 NEXT I
90 POKE32 &H000F0E80,BASE
100 POKE32 &H000F0E84,64
110 POKE32 &H000F0E90,11025
120 POKE32 &H000F0E94,255
130 POKE32 &H000F0E98,1
140 POKE32 &H000F0E9C,1
150 PRINT PEEK32(&H000F0E9C) AND 1
```

Expected result: the sample starts playing and line 150 prints 1 while the SFX channel is active. If the pointer or length is invalid, bit 1 is set instead.

For channels beyond 3, use the extended window. Channel n starts at $\$F2600 + n * \20 , so channel 4 begins at $\$F2680$. On the 6502 and Z80, select `IE_SFX_EXT_BANK` in bank 1 and use the $\$2600$ to $\$29FF$ aliases from the include file.

12.8 BASIC keyword map

Form	Effect
<code>SOUND ch, freq, vol[, wave[, duty]]</code>	Set frequency, volume, optional waveform, and optional duty for channel 0 to 3.
<code>ENVELOPE ch, atk, dec, sus, rel</code>	Set ADSR for channel 0 to 3.
<code>GATE ch, ON / GATE ch, OFF</code>	Start or release the channel envelope.
<code>SOUND WAVE ch, type</code>	Write channel <code>WAVE_TYPE</code> .
<code>SOUND NOISE ch, mode</code>	Write channel <code>NOISEMODE</code> .
<code>SOUND SWEEP ch, enable, period, shift</code>	Write channel sweep bits.
<code>SOUND SYNC ch, source</code>	Set hard-sync source.
<code>SOUND RINGMOD ch, source</code>	Set ring-modulation source.
<code>SOUND FILTER ...</code>	Global mixer filter, described in Chapter 11.
<code>SOUND REVERB ...</code>	Global reverb.
<code>SOUND OVERDRIVE ...</code>	Global overdrive.
<code>SOUND PLAY ... / SOUND STOP</code>	Media loader.

12.9 Side effects and limits

- `SOUND`, `ENVELOPE`, `GATE`, `SOUND WAVE`, `SOUND NOISE`, and `SOUND SWEEP` target channels 0 to 3.
- Direct `POKE32` reaches all ten channels.
- Direct SFX register writes reach all 32 extended SFX channels.

- Writing PHASE resets the oscillator phase.
- Writing DAC enables DAC mode for that channel; writing WAVE_TYPE returns the channel to oscillator mode.
- SFX_CTRL bit 0 triggers playback; bit 1 stops playback.
- SFX channel errors are reported in the SFX_CTRL read-back status.
- The legacy SFX window aliases only channels 0 to 3.
- Global overdrive, filter, and reverb are shared by all engines.

Chapter 13 covers the PSG.