nesdoug

14. Intro to Sound

Here's everything you never wanted to know about the APU, which makes the sounds on the NES. You can promptly forget everything once you start using a tool like Famitracker.

Go download this Sound Test NES file, written by SnoBrow

http://nesdev.com/sndtest.zip (http://nesdev.com/sndtest.zip)

(it may not work on every emulator, but it does work with FCEUX.)

Pressing 'select' switches sound channels. Pressing 'start' triggers whatever sound data you have on screen to play on that channel. In this program, Channel 0 = Square 1, Channel 1 = Square 2, Channel 3 = Triangle, Channel 4 = Noise.

The sound registers are mapped to $4000-4017.

4000-4003 = Square 1 registers

4004-4007 = Square 2 registers

4008-400b = Triangle registers

400c-400f = Noise registers

4010-4013 = DMC channel

4015 = controls audio output

4017 = frame counter

————————-

Terms borrowed from http://wiki.nesdev.com/w/index.php/APU (http://wiki.nesdev.com/w/index.php/APU)

4000-4003 = Square 1 channel

4000 = DDLC VVVV

D = duty cycle, changes the shape of the sound wave (basically, 10 = smooth, 01 or 11 = ok, 00 = annoying…reminds me of Atari sounds).

L = loop (repeats the sound)

C = constant volume

V = volume (sometimes)

If neither L or C set, it works like a volume envelope, that starts loud and gets quieter and quieter. If you set a note length of 0000 1 (in register 4003) you can really hear the volume envelope get quiter. V = length of note, V of 0 is still audible.

If L is set (and C not), V = how fast to repeat the sound. Smaller V = faster repeated sound. (Full volume) V of 0 is fast beeps.

If C is set (and not L), V controls the volume of the channel. The note will stop after a short period (length based on L bits in 4003). V of 0 is off.

If C and L set, V will control volume of the channel, and the note will play infinitely, until you write another value to 4000. This is actually how it's set all the time in music code. You control the length by counting frames, you make a volume envelope by changing the volume every frame, and end the note by either setting a volume of zero, or a frequency of zero. **To silence Square Channels, set a volume of zero ($4000 = $30).**

4001 – The frequency sweep register

4001 = EPPP NSSS

E = turns on a frequency sweep

P = smaller = faster sweep

N = 0 = sweep down, 1 = sweep up

S = smaller = faster sweep

Note, once it reaches the end of the sweep, the note will stop, even if you have it set to play constantly. Setting Sweep and Loop will repeat a few times until the sweep ends. This can produce cool effects.

4002 = TTTT TTTT = Low byte of the note frequency

4003 = LLLL LTTT

L = affects the length of the note, assuming you don't have it set to play constant notes (L & C both set in 4000). Very oddly set up. 0000 1 = a very long note.

T = high 3 bits of the note frequency.

Note: in order for for the lowest notes to play, the N flag ($4001) has to be set (for example, some games write $7f to $4001 when sweeps are off).

Also: the highest note playable is 000..0000 1000. Any frequency higher (such as 000..0000 0111) will produce silence from the Square 1 channel. You wouldn't want it any higher, it's very annoying at the upper frequency range.

— — — — — — — — — — —

4004-4007 = Square 2 Channel, exactly the same as Square 1.

— — — — — — — — — — — —

4008-400b = Triangle Channel

4008 – CRRR RRRR (C = constant, infinite note)

R = weird. Changing these numbers seems to effect the length of the note. Just set this to $ff for constant notes and $80 for off (silence). If set to $7f, the length will be controlled by the L bits of $400b. (Thanks **Pokun** for correcting me).

4009 – not used.

400a – TTTT TTTT – low byte of the note frequency

400b – LLLL LTTT – length and high bits of frequency…see the Sq 1 channel for details.

Note – There is no volume control for the Triangle Channel. Also, it will play 1 octave lower than the same settings on a square channel. Unlike the Square Channels, Triangle frequencies can go above 000 0000 1000 (such as 000 0000 0100), and play very very high pitches.

— — — — — — — — — — –

400c-400f = Noise Channel

400c – xxLC VVVV – same as Square Channel, except no duty cycle.

400d – not used

400e – Z- – – TTTT

Z = if 0, sounds like white noise

If 1, sounds like metallic clangs

T = note frequency, more = lower. A frequency of zero will play the highest pitch sound.

400f – LLLL L- – – –

L = affects the length of the note, assuming it isn't set to play constant notes (L & C both set).

To silence the Noise Channel, set a volume of zero ($400c = $30).

— — — — —

DMC channel, is a kind of low sample rate sound compression. The fastest sample rate gives a reasonable sound quality, but takes up a huge amount of ROM space. Slower sample rate takes up less space, but have an annoying high pitch hum, and lower quality sound.

Use DMC sparingly. You can also have very short looped samples, which might work for Bass Notes. I think the most common use for the DMC is improved drum sounds, and single words that sound like human speech – "Fight".

4010 – IL- – RRRR

I = set DMC triggered IRQs

L = Loop samples

R = Sample Rate (F = highest)

4011 – xDDD DDDD = Load. Essentially the starting value for a sample, some settings here will make a sample quieter. This can also be used (by continually changing the value) to make high quality PCM sound.

4012 – AAAA AAAA = Address of the sample…

%11AA AAAA.AA00 0000

Therefore, the lowest address is 11000000 00000000 = $C000

…and the highest address is 11111111 11000000 = $ffc0.

Note, DMC samples must be located between $c000 and $ffff. You can make samples with Famitracker.

4013 – LLLL LLLL = Length of the sample…

%LLLL.LLLL 0001

The smallest sample is 10001 = $11 bytes. The largest sample is $ff1 bytes. (You could play samples back to back, if that's too short).

4015 – Controls which channels are on. **USE THIS TO START AND STOP THE DMC CHANNEL.**

4015 – – – 5 4 3 2 1

1. Square 1

2. Square 2

3. Triangle

4. Noise

5. DMC

4017 – I never learned what this is for, except that the startup code for most games stores 0x40 here right at the start, I think to turn off some kind of IRQ.

— — — — — — —

For the first 4 Channels, the note will trigger when their last register is written…4003 triggers Square 1, 4007 triggers Square 2, 400b – Triangle, 400f Noise. Repeatedly writing to these (every frame) can produce unpleasant clicks.

To trigger a DMC sound, first write $0f to 4015, then write $1f to 4015. Or, if the DMC sample has ended, merely writing $1f to 4015 will retrigger the sample to play.

— — — — — —

Extra sound channels. Some advanced mappers (special cartridges) have additional sound channels. VRC7, for example has a more advanced FM synthesis chip (only used in 1 game, Lagrange Point).

— — — — — —-

PCM, ie. high quality sound. I've never tried this, but my understanding is, you can simulate high quality sound by writing values over and over to 4011. This would take so much of the CPU, that it would be impossible to run game logic while this was going on. Perhaps it could be done on a static Title Screen.

— — — — — — —-

Let's add some quick sound code, just for fun. Insert this into the infinite loop of one of the lessons, if you want. Make sure that sound channels are on…

((unsigned char)0x4015) = 0x0f;

//Quick beep –

if (((joypad1old & START) == 0)&&((joypad1 & START) != 0)){

((unsigned char)0x4000) = 0x0f;

((unsigned char)0x4003) = 0x01;

}

//sweep effect – jump sound

if (((joypad1old & START) == 0)&&((joypad1 & START) != 0)){

((unsigned char)0x4000) = 0x0f;

((unsigned char)0x4001) = 0xab;

((unsigned char)0x4003) = 0x01;

}

//Noise test –

if (((joypad1old & START) == 0)&&((joypad1 & START) != 0)){

((unsigned char)0x400c) = 0x0f;

((unsigned char)0x400e) = 0x0c;

```
*((unsigned char*)0x400e) = 0x00;

}
```

Feel free to play around with the SoundTest program, and then immediately, forget everything and move to Famitracker.

December 2, 2015August 25, 2017 dougfraker