

C=64 Mania – POKE cose sul SID

Giovanni A. Cignoni, Giuseppe Lettieri – Giugno 2015

Il *MOS 6581 Sound Interface Device*, per gli amici SID, è un *programmable sound generator*, cioè un circuito integrato capace di generare suoni interpretando dei valori numerici scritti in una serie di registri dedicati. I virtuosi del 64 combinano talento musicale e abilità di programmazione: di Rob Hubbard, per dire, oltre ai brani sono famose le routine assembler. Suonare il SID significa conoscere i suoi registri, dopodiché è “solo” questione di far mettere al 6510 i numeri giusti al tempo giusto. Molto spesso mentre fa altro, perché per esempio la musica è la colonna sonora di un gioco e il 6510 è per lo più impegnato con la gestione dello stato del gioco, la grafica e, va da sé, gli effetti sonori che alla musica di sottofondo si devono sovrapporre. E al tempo non c'erano, processi, thread, o processori multicore.

Il SID ha 25 registri in scrittura, mappati sulle locazioni da 54272 a 54296 della memoria del C=64. Giusto per informazione, ci sono anche 4 registri in lettura: 2 per ottenere i risultati dei due convertitori analogico-digitali e 2 per ottenere l'attuale valore degli 8 bit alti e dell'uscita e dell'involuppo dell'oscillatore 3.

I tre oscillatori del SID sono controllati ognuno da 7 registri. S=54272 è la base delle locazioni del SID.

Per l'oscillatore 1:

S+00 e S+01, definiscono la frequenza, byte basso e alto rispettivamente;

S+02 e S+03, controllano l'ampiezza dell'impulso per la forma d'onda a impulsi, byte basso e 4 bit alti;

S+04, ogni bit è interpretato a sé:

bit0, abilita il generatore d'involuppo (deve essere a 1 per sentire qualcosa);

bit1, sincronia con l'oscillatore 3 (per creare strutture armoniche particolari);

bit2, modulazione della forma d'onda triangolare con l'oscillatore 3 (per effetti di risonanza);

bit3, blocca l'oscillatore su 0 (per prove o effetti particolari);

bit4, seleziona la forma d'onda triangolare;

bit5, seleziona la forma d'onda a dente di sega;

bit6, seleziona la forma d'onda a impulso;

bit7, seleziona il segnale casuale (rumore);

S+05 e S+06, definiscono l'involuppo ADSR (attack/delay/sustain/release), 4 bit per ogni fase.

Gli oscillatori 2 e 3 sono controllati similmente dai registri da S+07 a S+13 e dai registri da S+14 a S+20.

I registri S+21 e S+22 controllano la frequenza di taglio del filtro, byte basso e 3 bit alti. Il registro S+23 controlla la risonanza (4 bit alti) e cosa passa dal filtro (4 bit bassi, i tre oscillatori e l'ingresso esterno). Il registro S+24 il modo di lavoro (4 bit alti, LP, BP, HP e 3off) e il volume di uscita (4bit bassi).

L'esempio proposto è minimale: suona solo una breve sequenza di note usando un oscillatore solo e in modo molto semplice. I valori per le frequenze delle note e per i parametri dell'involuppo ADSR fanno parte delle appendici del manuale utente del C=64.

Prove al volo

```
wf=16
s=54272
for i=s to s+24: poke i,0: next
poke s+5,9
poke s+6,76
poke s+2,128
poke s+24,15

poke s+1,28: poke s,49: poke s+4,wf+1
for i=0 to 255: poke s+1,i: next
for i=255 to 0 step -1: poke s+1,i: next
```

Esempio minimo in BASIC 64

```
10 wf=16
15 s=54272
20 for i=s to s+24: poke i,0: next
25 poke s+5,9
30 poke s+6,76
35 poke s+2,128
40 poke s+24,15
45 for t=1 to 1200: next
50 read hf, lf, d
60 if hf<0 then end
70 poke s+1,hf: poke s,lf
80 poke s+4,wf+1
85 for t=1 to d: next
90 poke s+4,wf: for t=1 to 24: next
100 goto 50
110 data 33, 135, 360
115 data 25, 30, 360
120 data 21, 31, 240
125 data 28, 49, 160
130 data 31, 165, 160
135 data 28, 49, 160
140 data 26, 156, 240
145 data 29, 223, 240
150 data 26, 156, 240
155 data 25, 30, 120
160 data 22, 96, 120
165 data 25, 30, 480
500 data -1, -1, -1
```