

# Scuola di videogame

## L'editor musicale: la routine IRQ

*Sorpresi? Niente paura il «videogioco tutto nostro» ha solo preso una puntata di ferie, almeno dal punto di vista del titolo iniziale, visto che in questo numero le due sezioni del Megagame 64 sono fuse insieme. Infatti parleremo del listato che trovate in queste pagine che, oltre ad essere la routine fondamentale dell'editor musicale delle scorse due puntate, sarà inserita nel megagame per realizzare la colonna sonora*

Primo avvertimento da farvi è: la routine è perfettamente (...forse sono stato troppo ottimista) funzionante, ma manca di alcune parti, come la modulazione del DUTY-CICLE e gli effetti «speciali», tuttavia può essere utilizzata anche così come ve la presento; in seguito vi comunicherò le aggiunte da fare... sempre che ciò vi interessi. Cerchiamo di non creare confusione e cominciamo dall'inizio.

Innanzitutto vediamo come è stato scritto il listato, ovvero quale Assembler ho usato; il suo nome, almeno a detta della scritta di partenza, è Assembler V3 di Yves Han. Si lancia con un SYS 4408 e lascia a disposizione dell'utente 36324 byte liberi. È un Assembler molto pratico perché fa diretto utilizzo dell'editor del Basic, ovvero possiamo utilizzare istruzioni in LM come se fossero istruzioni Basic. È possibile quindi utilizzare variabili per le etichette e altri comandi come l'OLD o l'AUTO che ci permettono, rispettivamente, di recuperare un programma cancellato con NEW e di avere la numerazione automatica delle linee del listato, ad ogni return; altre possibilità sono l'utilizzo di nomi per variabili ed etichette lunghi fino ad 8 caratteri e di cifre in esadecimale (precedute da \$) o binarie (precedute da %). Se non possedete un Assembler di questo tipo la digitazione del listato si presenta molto tediosa.

Le linee dalla 100 alla 120 sono le direttive per assemblare tutto ciò che si trova tra queste e il comando NEXT PASS (linea 14000).

La routine si avvia, una volta assemblata, con SYS 28672 (\$7000) e si «accoda» ad eventuali modifiche della IRQ, visto che, una volta terminata la sua esecuzione, salta al valore precedentemente contenuto nelle locazioni \$0314 e \$0315 e non direttamente a \$EA31.

Dalla linea 200 in poi troviamo la routine musicale vera e propria. Nella locazione indicata dall'etichetta MUSATT (d'ora in avanti diremo semplicemente il nome dell'etichetta) bisogna depositare un valore da 1 a 10, ovvero scegliere la musica da attivare. Lo zero

equivale a nessuna musica e in tal caso si salta alla OLDIRQ.

Depositato un numero valido (la routine non controlla se il numero inserito è maggiore di 10) nel caso esso sia differente dall'ultima IRQ (UMUSATT indica il valore che aveva la MUSATT all'IRQ precedente) si attiva la nuova musica saltando a NEWMUS. Si codifica quindi il valore fornito in pezzo iniziale e pezzo finale, servendosi della tabella MUSICA che è stata precedentemente ordinata dall'utente con una serie di 20 valori (o meno, se le musiche sono minori di 10), che indicano appunto il pezzo iniziale e il pezzo finale di ogni musica.

L'assegnazione dei valori alle LABEL del listato avviene a partire dalla linea 13000. Decodificati il PPEZZO e il CPEZZI si passa all'identificazione della traccia di start (STRACC) e del numero di tracce da eseguire (ETRACC) e si effettua la riassociazione dei 16 strumenti. Vediamo come deve essere preparata questa struttura dati.

A partire dalla locazione DATI e per le 99 successive troviamo tutte le STRACC; da DATI+100 a DATI+199 tutte le ETRACC; da DATI+200 a DATI+299 tutti i primi e secondi strumenti (rispettivamente nel nibble basso e nel nibble alto) e così via fino al 15esimo e 16esimo strumento. Questo per ciò che riguarda i pezzi. Con la STRACC si ricavano i tre blocchi delle voci che sono disposti nel seguente modo: da DATI+1000 a DATI+1255 tutti i blocchi delle voci 1; da DATI+1256 a DATI+1511 quelli della voce 2; da DATI+1512 a DATI+1767 quelli della terza voce. Impostata la CPAUSA ad 1, che, essendo un contatore per attendere la IRQ dell'aggiornamento dati, viene decrementata ad ogni IRQ, l'aggiornamento sarà immediato. La VPAUSA è il valore da assegnare a CPAUSA (selezionabile dall'utente) quando quest'ultimo si azzerà. Dalla linea 750 in poi troviamo l'aggiornamento della prima voce. Esso avviene solo nel caso la voce sia attiva, ovvero se in VOCEATT1 c'è uno zero (per le voci 2 e 3 si controlla VOCEATT2 e VOCEATT3). Normalmente tutte le

voci sono attive. A questo punto bisogna ricavare la locazione di memoria dalla quale prelevare la nota e lo strum/eff, basandosi sul PBLOC1 (numero del blocco della voce 1). I blocchi sono disposti a partire dalla locazione DATI+1792; il primo byte di un blocco indica la nota (da 0 a 93, per coprire le otto ottave disponibili), mentre il secondo indica lo strum/eff e così via fino al 128esimo byte. Per ogni unità indicata dal PBLOC1 quindi occorre un incremento della locazione base di 128 byte. Ciò si ottiene con la seguente operazione: si prende il PBLOC1 e lo si shifta (LSR) a destra; se il CARRY è settato vuol dire che il PBLOC1 contiene un valore dispari quindi sicuramente ci deve essere un delta dalla locazione DATI+1792 di 128 nel byte basso del puntatore che stiamo costruendo, quindi in \$FB viene depositato 128 e il valore contenuto nell'accumulatore in seguito allo SHIFT viene usato come byte alto, quindi depositato in \$FC. Ora in \$FB, \$FC c'è il delta da aggiungere a \$FC il DATI+1792/256 che essendo allineato con la «pagina» non comporta resto da aggiungere a \$FB. Trasferiamo nel registro Y il valore contenuto in CBLOC (puntatore interno del blocco) e usiamo l'indirizzamento indicizzato per

ottenere finalmente il valore della nota. Nell'eventualità che detto valore sia il 254 la routine lo riconosce come STOP, quindi si organizza in modo da farlo riconoscere come tale alla prossima routine di gestione delle voci; se invece il valore è il 255 allora si tratta di uno SPAZIO e la voce non deve subire aggiornamenti (si passa direttamente a quella successiva). Se infine il valore è compreso tra 0 e 93 (gli altri valori non devono essere utilizzati) vengono ricavati i LB e HB della frequenza della nota tramite dedicate tabelle (NOTAL, NOTAH). Ottenuta la nota si preleva il successivo byte che viene scomposto in strumento e in effetto. Un comportamento simile lo abbiamo per le altre due voci fin quando non arriviamo alla routine CKCONT, la quale serve al fine di aggiornare il CBLOC e eventualmente la STRACC, quindi il PPEZZO ecc. Se la musica è terminata si ricomincia da capo.

Siamo giunti così alla routine STRUMEFF che è appunto quella che si occupa degli aggiornamenti delle voci e della gestione delle modulazioni (...in un prossimo futuro anche degli effetti).

Vediamo la struttura dati degli strumenti. Dalla linea 13530 in poi troviamo 16 blocchi di dati che settano i 16

strumenti con parametri scelti a caso. Il primo byte di ogni blocco dati contiene l'AD, segue quindi l'SR, la forma d'onda, il DUTY-CICLE (2 byte), la frequenza di taglio del filtro (2 byte), la risonanza, i filtri da attivare, la frequenza di modulazione del filtro, quella di modulazione della frequenza-nota e quella di modulazione del duty-cycle (anche se, per ora, non è utilizzata), segue ancora l'ampiezza della modulazione del filtro, quella della frequenza e quella del duty-cycle. Il 16esimo byte è di riempimento. La struttura dati EFFDATI non è presente ma dovrebbe comprendere un blocco di 10 byte per ogni effetto. Per ogni voce ci sono due buffer che contengono lo strumento e l'effetto attuali, o meglio tutti i dati della struttura che li identifica. Vediamo come dal valore contenuto in STRUM1 (o comunque STRUM2 o STRUM3), che è compreso tra 0 e 15, si ricavano i dati da inserire nel buffer (BUFSTR1, 2 o 3). Sappiamo già che in STRUM1 non c'è il numero dello strumento da utilizzare come effettivo, ma che tale numero deve essere convertito servendosi della tabella realizzata dalla routine che seleziona i pezzi (quella vista nella parte relativa all'impostazione iniziale della musica). Ottenuto il numero reale dello strumento possiamo final-

```

100 FOR PASS=1 TO 3
110 ORG $7000
120 IF PASS=3 THEN ORG $7000,1
125 DATI=$7000+10*256
130 BE1:LDA #0313:STA OLDIRG+1
140 LDA #PARCO/256
150 STA #0313:LDA #0314:STA OLDIRG
160 LDA #((PARCO/256)-INT(PARCO/256))*256
170 STA #0314
180 DLI
190 RTS
200 PARCO: ROUTINE MUSICALE
210 LDA UMUSATT
220 CMP MUSATT
230 BNE NEMUS; E' UNA NUOVA MUSICA
240 CMP #0
250 BEQ OLDI;SE LA MUSICA NON E' NULLA SALTA AL BRANO
255 JMP ROUTMUS
260 OLDI:JMP (OLDIRG)
270 NEMUS:
280 LDA MUSATT
290 STA UMUSATT:DEC MUSATT:LDA MUSATT
300 ASL
310 TAX:INC MUSATT
320 LDA MUSICA,X
330 STA PPEZZO
340 INX
350 LDA MUSICA,X
360 STA CPEZZI:JSR AGGIORNO:JMP FINEAGG
370 AGGIORNO:LDX PPEZZO
380 LDA DATI,X
390 STA STRACC
400 LDA DATI+100,X
410 STA ETRACC
420 LDA DATI+200,X
430 STA STRUM
440 LDA DATI+300,X
450 STA STRUM+1
460 LDA DATI+400,X
470 STA STRUM+2
480 LDA DATI+500,X
490 STA STRUM+3
500 LDA DATI+600,X
510 STA STRUM+4
520 LDA DATI+700,X
530 STA STRUM+5
540 LDA DATI+800,X
550 STA STRUM+6
560 LDA DATI+900,X
570 STA STRUM+7
580 AGSTR:LDX STRACC
590 LDA DATI+1000,X
600 STA PBLOC1
610 LDA DATI+1256,X
620 STA PBLOC2
630 LDA DATI+1512,X
640 STA PBLOC3
650 LDA #0
660 STA CBLOC:RTS:FINE ROUT AGGIORNAM.
670 FINEAGG:LDA #1
680 STA CPAUSA
690 ROUTMUS
700 LDX CPAUSA
710 DEX
720 BEQ AZZ1; FINE DELLA PAUSA
725 STX CPAUSA:JMP STRUMEFF
730 AZZ1:LDA VPAUSA:QUINDI AGGIORNA LE
740 STA CPAUSA
750 CLC:LDA #00:STA #FB
755 LDA VOCEATT1:BEQ KK1:JMP VOCE2
760 KK1:LDA PBLOC1
770 LSR
780 BEC SALLI:NIENTE INCREMENTO DI 128
790 STA #FC:LDA #128:STA #FB
810 JMP CIKK1
820 SALL1:STA #FC
840 CIKK1:X=(DATI+1792)/256:CLC:LDA #X
850 ADC #FC:STA #FC
860 LDY CBLOC
870 LDA (#FB),Y:IN A C'E' LA NOTA
880 TAX

```

(continua a pag. 168)

(segue da pag. 167)

```

2 883 CMP #254:BEQ ESTOP:SETTA LO STOP
885 CMP #255:BEQ SPAZIO:E' UNO SPAZIO
887 JMP SAL2
888 ESTOP:LDA #1:STA RELEA1:JMP VOCE2
889 SPAZIO:LDA #0:STA ATTIVA1:JMP VOCE2
890 STA FRQ1
910 LDA NOTAH,X
920 STA FRQ1+1
930 INV
940 LDA (#FB),Y:ADESSO DECODIFICA
950 AND #15
960 STA STRUM1
970 LDA (#FB),Y
980 LSR:LSR:LSR
990 STA EFF1:LDA #1:STA ATTVA1
991 LDA BUFSTR+2:AND #254:STA #D404:JMP VOCE2
995 JMP VOCE2
1000 CKCONT:LDY #00:JSR CKCK:TYA:BNE RMSK:JMP STRUMEFF:AGGIORNA I PUNTORI
1005 RMSK:JMP NEMMUS
1010 CKCK:INC CRLOC:INC CBLOC:LDA CBLOC
1020 CMP #128
103C BEQ VAL1
1040 RTS
1050 VAL1
1060 INC STRACC:DEC ETRACC
1070 LDA ETRACC
1080 CMP #255
1090 BEQ APEZ
1100 JSR ABSTR
1110 RTS
1120 APEZ:JMP DEC CPEZZI
1130 INC PREZIO:DEC CPEZZI
1140 LDA CPEZZI
1150 CMP #255
1160 BEQ REPUBIC
1170 JSR ABGIORND
1180 RTS
1190 REMUSIC:LDY #FF:RTS
1730 VOCE2:CLC:LDA #00:STA #FB
1735 LDA VOCEAT:BEQ KK2:JMP VOCE3
1740 LK2:LDA PBLOC2
1750 LK2:SSAL1:NIENTE INCREMENTO DI 128
1760 BCC SSAL1:NIENTE INCREMENTO DI 128
1790 STA #FC:LDA #128:STA #FB
1810 JMP CIKK2
1820 SSAL1: STA #FC
1830 ORA IN #FB:#FC:C'E' IL DELTA
1840 CIKK2:X=(DAT1+1792)/256:CLC:LDA #X
1850 ADC #FC:STA #FC
1860 LDY CBLOC
1870 LDA (#FB),Y:IN A C'E' LA NOTA
1880 TAX
1883 CMP #254:BEQ SESTOP:SETTA LO STOP
1885 CMP #255:BEQ SESPAZIO:E' UNO SPAZIO
1887 JMP SSAL2
1888 SESTOP:LDA #1:STA RELEA2:JMP VOCE3
1889 SESPAZIO:LDA #0:STA ATTIVA2:JMP VOCE3
1890 SSAL2:LDA NOTAH,X
1900 STA FRQ2
1910 LDA NOTAH,X
1920 STA FRQ2+1
1930 INV
1940 LDA (#FB),Y:ADESSO DECODIFICA
1950 AND #15
1960 STA STRUM2
1970 LDA (#FB),Y
1980 LSR:LSR:LSR
1990 STA EFF2:LDA #1:STA ATTVA2
1991 LDA BUFSTR+2:AND #254:STA #D400+11:JMP VOCE3
2750 VOCE3:CLC:LDA #00:STA #FB
2755 LDA VOCEATT3:BEQ KK3:JMP CKCONT
2760 KK3:LDA PBLOC3
2770 LSR
2780 BCC SSSAL1:NIENTE INCREMENTO DI 128
2790 STA #FC:LDA #128:STA #FB
2810 JMP CIKK3
2820 SSSAL1: STA #FC
2830 ORA IN #FB:#FC:C'E' IL DELTA
2840 CIKK3:X=(DAT1+1792)/256:CLC:LDA #X
2850 ADC #FC:STA #FC

```

LO STRUMENTO E L'EFFETTO

LO STRUMENTO E L'EFFETTO

4

```

4027 SEVOCE2
4028 LDA VOCEATT2:BEQ VK2:JMP SEVOCE3
4029 VK2:JSR VH2:JMP SEVOCE3:VH2:LDA ATTIVA2
4030 BNE TZE5T1
4040 LDA RELEA2
4050 BNE TZE5T2
4060 RTS
4070 TZE5T1:JMP AZTTVOCE
4080 TZE5T2
4090 LDA BUFSTR2+2
4100 AND #254
4110 STA #D400+11:LDA #0:STA RELEA2
4120 RTS
4130 AZTTVOCE
4140 LDA STRUM2:LDY #0
4141 CLC:LSR:BCC O2P11:LDY #1
4150 O2P11:TAX:LDA STRUM,X
4155 DEV:BEQ T2AXIT
4160 ASL:ASL:ASL:ASL
4170 TAX:JMP L2P1
4175 T2AXIT:AND #240:TAX
4176 L2P1:LDY #0
4180 L200P1:LDA STRDAT1,X
4190 STA BUFSTR2,Y
4200 INX:INX:CPY #15
4210 BNE L200P1
4220 LDA EFF2
4230 ASL:ASL:ASL:ADC EFF2:ADC EFF2
4240 TAX:LDY #0
4250 L200P2:LDA EFFDAT1,X
4260 STA BUFEFF2,Y
4270 INX:INX:CPY #10
4280 BNE L200P2
4290 LDA #0:STA ATTIVA2:STA FLDIR2:STA NINCC
4300 LDA BUFSTR2
4310 STA #D400+12:AD
4320 LDA BUFSTR2+1
4330 STA #D400+13:SR
4340 LDA BUFSTR2+3
4350 STA #D400+9: D-C LOW
4360 LDA BUFSTR2+4
4370 STA #D400+10: D-C HIGH
4380 LDA BUFSTR2+5
4390 BNE N2SALTAR
4400 LDA BUFSTR2+6
4410 BEQ 2SALTAR5
4420 N2SALTAR5:STA #D400+22:STA FFIL+1
4430 LDA BUFSTR2+5:STA #D00+21:STA FFIL
4445 LDA BUFSTR2+6:STA AMPFFIL
4470 ORA #2:STA #D400+23:STA RLT:JMP R2LT2
4480 2SALTAR5:LDA RLT:AND #253:STA RLT:STA #D400+23
4490 R2LT2:LDA BUFSTR2+B
4500 ORA VOLUME
4510 STA #D400+24
4520 LDA FRQ2
4530 STA #D400+7
4540 LDA FRQ2+1
4550 STA #D400+8
4570 LDA BUFSTR2+2:STA #D400+11:RTS
5027 SEVOCE3
5028 LDA VOCEATT3:BEQ VK3:JMP STREFF
5029 VK3:JSR VH3:JMP STREFF:VH3:LDA ATTIVA3
5030 BNE TZE5T1
5040 LDA RELEA2
5050 BNE TZE5T2
5060 RTS
5070 TZE5T1:JMP A3TTVOCE
5080 TZE5T2
5090 LDA BUFSTR3+2
5100 AND #254
5110 STA #D400+18:LDA #0:STA RELEA3
5120 RTS
5130 A3TTVOCE
5140 LDA STRUM3:LDY #0
5141 CLC:LSR:BCC O3P11:LDY #1
5150 O3P11:TAX:LDA STRUM,X
5155 DEV:BEQ T3AXIT
5160 ASL:ASL:ASL:ASL
5170 TAX:JMP L3P1
5175 T3AXIT:AND #240:TAX

```

3

```

2860 LDY C80C
2870 LDA (#FB),Y;IN A C'E' LA NOTA
2880 TAX
2890 CMP #254;BEG SSESTOP;SETTA LD STOP
28B3 CMP #255;BEG SESPAAZIO;E' UNO SPAZIO
28B7 JMP SSSAL2
28B8 SSESTOP;LDA #1;STA RELEA3;JMP CKCONT
28B9 SESPAAZIO;LDA #0;STA ATTIVAZ3;JMP CKCONT
2890 SSSAL2;LDA NOTAH,X
2900 STA FRQ3
2910 LDA NOTAH,X
2920 STA FRQ3+1
2930 INY
2940 LDA (#FB),Y;ADESSO DECODIFICA
2950 AND #15
2960 STA STRUM3
2970 LDA (#FB),Y
2980 LSR;LSR;LSR;LSR
2990 STA EFF3;LDA #1;STA ATTIVAZ
2991 LDA BUFSTR3+2;AND #254;STA #D400+18
3000 JMP CKCONT
3027 STRUMEFF
3028 LDA VOCEATT1;BEG VK1;JMP SEVOICE2
3029 VK1;JMK VHI;JMP SEVOICE2;VHI;LDA ATTIVAZ1
3030 BNE TEST1
3040 LDA RELEA1
3050 BNE TEST2
3060 RTS
3090 TEST1;JMP ATTVOICE
3100 AND #254
3110 STA #D400+4;LDA #0;STA RELEA1
3120 RTS
3130 ATTVOICE;TRASFERISCE UN NUOVO STR
3140 LDA STRUM1;LDA #0
3141 CLC;LSR;BCC;JPH;LDY #1
3150 DEY;BEG TAXIT;STRUM,X
3155 DEY;BEG TAXIT;STRUM,X
3160 ASL;ASL;ASL;ASL;MOLTIPL. PER 16
3170 TAX;JMP LPA
3175 TAXIT;AND #240;TAX
3176 FLDY;LDY #0
3180 LDD;FILD;LDA STRDAT1,X
3190 STA BUFSTR,Y
3200 INX;INX;CPY #15
3210 BNE LOOP1
3240 TAX;LDY #0
3250 LDD;FILD;LDA EFFDAT1,X
3260 STA BUFEFF,Y
3270 INX;INX;CPY #10
3280 BNE LOOP2
3290 LDA #0;STA ATTIVAZ1;STA FLDIR;STA NINC
3300 LDA BUFSTR
3310 STA #D400+5; AD
3320 LDA BUFSTR+1
3330 STA #D400+6; SR
3340 LDA BUFSTR+3
3350 STA #D400+2; D-C LOW
3360 LDA BUFSTR+4
3370 STA #D400+3; D-C HIGH
3380 LDA BUFSTR+5
3390 BNE NSALTAF
3400 LDA BUFSTR+6
3410 BEQ S3ALR15
3420 NSALTAF;STA #D400+22;STA FFIL+1
3430 LDA BUFSTR+5;STA #D400+21;STA FFIL
3440 LDA BUFSTR+12;STA AMPFIL
3450 LDA BUFSTR+9;STA FRQFIL
3450 LDA BUFSTR+7;ORA RL
3470 ORA #1;STA #D400+23;STA RL;JMP RL2
3480 S3ALR15;LDA RL;AND #254;STA RL;STA #D400+23
3490 RL2;LDA BUFSTR+8
3500 ORA VOLUME
3510 STA #D400+24
3520 LDA FRQ1
3530 STA #D400
3540 LDA FRQ1+1
3550 STA #D401
3570 LDA BUFSTR+2;STA #D404;RTS ;
    
```

E INFINE

^SETTA LA FORMA D'ONDA

5

```

5176 L3P;LDY #0
5180 L300P1;LDA STRDAT1,X
5190 STA BUFSTR3,Y
5200 INX;INX;CPY #15
5210 BNE L300P1
5220 LDA EFF3
5230 ASL;ASL;ASL;ADC EFF3;ADC EFF3
5240 TAX;LDY #0
5250 L300P2;LDA EFFDAT1,X
5260 STA BUFEFF3,Y
5270 INX;INX;CPY #10
5280 BNE L300P2
5290 LDA #0;STA ATTIVAZ3;STA FLDIR3;STA NINC3
5300 LDA BUFSTR3
5310 STA #D400+19; AD
5320 LDA BUFSTR3+1
5330 STA #D400+20; SR
5340 LDA BUFSTR3+3
5350 STA #D400+16; D-C LOW
5360 LDA BUFSTR3+4
5370 STA #D400+17; D-C HIGH
5380 LDA BUFSTR3+5
5390 BNE NSALTAF
5400 LDA BUFSTR3+6
5410 BEQ S3ALR15
5420 NSALTAF;STA #D400+22;STA FFIL+1
5430 LDA BUFSTR3+5;STA #D400+21;STA FFIL
5440 LDA BUFSTR3+12;STA AMPFIL
5445 LDA BUFSTR3+9;STA FRQFIL
5450 ORA #4;STA #D400+23;STA RL;JMP RL2
5470 S3ALR15;LDA RL;AND #251;STA RL;STA #D400+23
5490 RL2;LDA BUFSTR3+8
5500 ORA VOLUME
5510 STA #D400+24
5520 LDA FRQ3
5530 STA #D400+14
5540 LDA FRQ3+1
5550 STA #D400+15
5570 LDA BUFSTR3+2;STA #D400+18;RTS
6000 STREFF;MODULAZIONE DI FREQUENZA
6005 LDY #0;LDY #0
6010 STRLOOP;LDA FLDIR,X
6020 BNE RSBC
6030 LDA FRQ1+1,X
6040 CMP #255
6050 BNE INCREM
6060 CLC;LDA FRQ1,X
6070 ADC BUFSTR+13;Y
6080 BCC INCREM
6090 LDA #1
6100 STA FLDIR,X
6110 JMP STREFF2
6120 INCREM;CLC;LDA FRQ1,X
6130 ADC BUFSTR+13;Y;STA FRQ1,X
6140 BCC NONINC
6150 INC FRQ1+1,X
6160 NONINC;INC NINC,X
6170 LDA NINC,X
6180 CMP BUFSTR+10;Y
6190 BEQ SONOUG
6200 JMP STREFF2
6210 SONOUG;LDA #1
6220 STA FLDIR,X
6230 JMP STREFF2
6240 RSBC
6250 LDA FRQ1+1,X
6260 CMP #0
6270 BNE INCR2
6280 CLC;LDA FRQ1,X
6290 SBC BUFSTR+13;Y
6300 BCS INCR2
6310 LDA #0
6320 STA FLDIR,X
6330 JMP STREFF2
6340 INCR2;CLC;LDA FRQ1,X
6350 SBC BUFSTR+13;Y;STA FRQ1,X
6360 BCS NONDEC
6370 DEC FRQ1+1,X
6380 NONDEC
6390 DEC NINC,X
6400 CLC;LDA #0;SBC BUFSTR+10;Y
    
```

(continua a pag. 170)

(segue da pag. 169)

6

```

6420 CMP NINC,X
6430 BEQ INVERT
6440 JMP STREFF2
6450 INVERT:LDA #0
6460 STA FLDIR,X
6500 STREFF2
6510 INX:INX:TYA:CLC:ADC #25:YAY
6520 CPX #6:BEQ STREFF3
7000 STREFF3:MODULAZIONE DEL FILTRO
7010 LDA F2DIR
7020 BNE ARSBC
7030 LDA FFIL+1
7040 CMP #7
7050 BNE AINCREM
7060 CLC:LDA FFIL
7070 ADC AMPFIL
7080 BCC AINCREM
7090 LDA #1
7100 STA F2DIR
7110 JMP STREFF4
7120 AINCREM:CLC:LDA FFIL
7130 ADC AMPFILLESTA FFIL
7140 BCC ANONINC
7150 INC FFIL+1
7160 ANONINC:INC NZINC
7170 LDA NZINC
7180 CMP FRGFIIL
7190 BEQ ARSBCDUG
7200 JMP STREFF4
7210 ARSBCDUG:LDA #1
7220 STA F2DIR
7230 JMP STREFF4
7240 ARSBC
7250 CMP #6:FFIL+1
7260 BNE AINCR2
7270 CLC:LDA FFIL
7280 BCC AMPFIL
7290 BCS AINCR2
7300 LDA #0
7310 STA F2DIR
7320 JMP STREFF4
7330 AINCR2:CLC:LDA FFIL
7340 SBC AMPFILLESTA FFIL
7350 BCC ANONDEC
7360 DEC FFIL+1
7370 ANONDEC
7380 DEC NZINC
7400 CLC:LDA #0:SBC FRGFIIL
7420 CMP NZINC
7430 BEQ AINVERT
7440 JMP STREFF4
7450 AINVERT:LDA #0
7460 STA F2DIR
7500 STREFF4
9070 MODULA
9080 LDA FRG1:STA $D400:LDA FRG1+1:STA $D401
9090 LDA FRG2:STA $D400+7:LDA FRG2+1:STA $D401+7
9100 LDA FRG3:STA $D400+14:LDA FRG3+1:STA $D401+14
9180 LDA FFIL:STA $D400+21:LDA FFIL+1:STA $D400+22
10000 JMP (OLDIRG)
13000 NUMMUS: BYTE 0
13010 VOCEATT1:BYTE 0:VOCEATT2:BYTE 0:VOCEATT3:BYTE 0
13020 MUSATT: BYTE 0
13030 UMUSATT:BYTE 0
13040 PPEZZO: BYTE 0
13050 CPEZZI: BYTE 0
13060 STRACC: BYTE 0
13070 ETRACC: BYTE 0
13080 STRUM: BYTE 0,0,0,0,0,0,0,0
13090 PBLCC1: BYTE 0
13100 PBLCC2: BYTE 0
13110 PBLCC3: BYTE 0
13115 CBLOC: BYTE 0
13120 CPAUSA: BYTE 10
13125 VPAUSA: BYTE 10
13126 FLDIR:BYTE 0:NINC:BYTE 0
13127 FLDIR2:BYTE 0:NINC2:BYTE 0
13128 FLDIR3:BYTE 0:NINC3:BYTE 0
13129 F2DIR:BYTE 0:N2INC:BYTE 0
13130 MUSICA:

```

7

```

13140 BYTE 0,1,0,40,0,60,8,19,0,0
13145 BYTE 0,0,0,0,0,0,0,0,0,0,0
13150 FRQ1: WORD 0
13160 FRQ2: WORD 0
13170 FRQ3: WORD 0
13175 FFIL: WORD 0
13176 AMPFIL:BYTE 0:FRGFIIL:BYTE 0
13180 RELEA1: BYTE 0
13190 RELEA2: BYTE 0
13200 RELEA3: BYTE 0
13220 OLDIRG: WORD 0
13230 ATTIVA1: BYTE 0
13240 ATTIVA2: BYTE 0
13250 ATTIVA3: BYTE 0
13260 STRUM1: BYTE 0
13270 STRUM2: BYTE 0
13280 STRUM3: BYTE 0
13290 EFF1: BYTE 0
13300 EFF2: BYTE 0
13310 EFF3: BYTE 0
13315 RLT: BYTE 0
13316 VOLUME: BYTE 15
13320 BUFSR: BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
13330 BUFSR2:BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
13350 BUFSR2:BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
13360 BUFSR2:BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
13370 BUFSR3:BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
13380 BUFSR3:BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
13390 NOTAH:BYTE 22,39,57,75,95,116,138,161,186,212,240,14
13400 BYTE 45,78,113,150,190,231,20,66,116,169,224,27,90,156,226
13410 BYTE 45,123,207,39,133,232,81,193,55,180,56
13420 BYTE 196,89,247,158,78,10,208,162,129,109,103,112
13430 BYTE 137,178,237,59,157,20,160,69,3,219
13440 BYTE 207,225,18,101,219,118,58,39,65,138,5,181,157
13450 BYTE 193,36,201,182,237,115,78,130,20,10,106,59
13460 BYTE 130,72,147,107,218,231,156,4,40,20
13470 NOTAH:BYTE 1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2
13480 BYTE 3,3,3,3,3,4,4,4,4,5,5,5,6,6,7,7,8,8,9,9,10,10
13490 BYTE 11,12,13,14,15,16,17,18,19,20,21,23,24,26,27
13500 BYTE 29,31,32,34,36,39,41,43,46,49,52,55,58,62,65,69,73,78,82,87,92,98
13510 BYTE 104,110,117,124,131,139,147,156,165,175,185
13520 BYTE 196,208,221,234,248
13530 STRDAT1
13540 BYTE $0A,04,65:00,04,00,00,240,00,02,1,20,255,0,255,0
13550 BYTE $2C,04,17,10,10,00,00,240,0,2,2,2,10,10,100,0
13560 BYTE $2F,04,65,04,04,30,50,240,32,10,10,10,10,10,0
13570 BYTE $2F,04,129,10,04,00,00,240,0,2,1,2,10,10,255,10,0
13580 BYTE $72,04,33,00,00,00,00,240,0,2,2,2,10,10,10,0
13590 BYTE $09,04,17,00,00,00,00,240,0,2,2,2,10,10,10,0
13600 BYTE $07,04,33,00,00,80,10,240,32,10,10,10,250,10,0
13610 BYTE $29,04,17,00,00,00,00,240,0,2,2,2,10,10,10,0
13620 BYTE $29,04,33,00,00,00,00,240,0,2,2,2,10,10,10,0
13630 BYTE $09,04,17,00,00,00,00,240,0,2,2,2,10,10,10,0
13640 BYTE $09,04,33,00,00,00,00,240,0,2,2,2,10,10,10,0
13650 BYTE $09,04,33,00,00,00,00,240,0,2,2,2,10,10,10,0
13660 BYTE $09,04,33,00,00,00,00,240,0,2,2,2,10,10,10,0
13670 BYTE $09,04,33,00,00,00,00,240,0,2,2,2,10,10,10,0
13680 BYTE $09,04,33,00,00,00,00,240,0,2,2,2,10,10,10,0
13690 BYTE $09,00,00,00,00,00,00,240,0,2,2,2,10,10,10,0
13700 BYTE $09,00,00,00,00,00,00,240,0,2,2,2,10,10,10,0
14000 NEXT PASS
15000 POKE DATI+1,0
15020 POKE DATI+100,255
15030 POKE DATI+200,0
15040 POKE DATI+300,1+16
15050 POKE DATI+400,2+32
15060 POKE DATI+500,3+48
15070 POKE DATI+600,4+64
15080 POKE DATI+700,5+80
15090 POKE DATI+800,6+96
15100 POKE DATI+900,7+112

```

READY.

# Megaposta

Ragazzi, sono arrivate diverse lettere in risposta al mio invito di scrivere una storia per il megagame! Sono molto contento per questo, ma non posso cominciare a pubblicarne i contenuti, sia perché questa puntata è già abbastanza pienotta, sia perché voglio aspettare ancora un po'. Chissà che non ne venga fuori un nuovo Asimov...

mente trasferire con un loop (linee 3180-3210) di 16 passate i dati nel buffer. Segue la routine di trasferimento dei dati dell'effetto che utilizza il suo loop a 10 passate (linee 3250-3280). Ottenuti i nostri buffer possiamo effettuare il trasferimento decisivo nei registri del SID, dedicando particolare attenzione all'utilizzo dei filtri (linee 3380-3510). L'ultimo parametro è ovviamente la forma d'onda, che come noto fa «partire» il processore. Andiamo ad analizzare la subroutine STREFF, la quale si occupa delle modulazioni (linee da 6000 in poi). La prima modulazione è quella della fre-

quenza; questa parte di routine viene eseguita 3 volte per le tre voci. La «variabile» FLDIR indica la direzione, ovvero se la frequenza deve essere incrementata o decrementata. La variabile NINC indica il numero di volte che la frequenza è stata INC/DECREMENTATA. Il valore dell'AMPIEZZA della modulazione viene addizionato/sottratto fin quando il valore di NINC diventa pari alla FREQUENZA di modulazione (quando si decrementa il valore deve essere pari al negativo della frequenza); a tal punto si inverte il valore di FLDIR (da 0 a 1 o viceversa). Nell'eventualità che l'INC/

DECREMENTO comporti un'uscita dai limiti imposti il FLDIR verrà invertito prima di ottenere una... stonatura (in altre parole, se, ad esempio, un incremento porterebbe la frequenza della nota ad un valore più alto del massimo, non ci sarà incremento, quindi non ci sarà il passaggio alle basse frequenze). Anche per il filtro c'è la modulazione, che in questo caso è unica per le tre voci e inoltre ha un range più ristretto. La routine MODULA (9070 e seguenti) trasferisce i valori della frequenza e del taglio del filtro nei rispettivi registri SID. Da notare che se non si desidera avere modulazione basta impostare l'AMPIEZZA pari a zero. Il volume del sonoro deve essere impostato nella locazione VOLUME e non direttamente nel SID, perché altrimenti si andrebbero ad alterare i filtri impostati dalla routine. Le linee della 15010 in poi servono a creare una struttura dati iniziale per ciò che riguarda i pezzi da suonare, ma non è indispensabile (l'ho usata io per le prove). Non mi resta altro da aggiungere se non che nella prossima puntata ci occuperemo dell'editor (speriamo di poter pubblicare il listato) e delle parti mancanti. A presto.

IMPORTAZIONE E DISTRIBUZIONE DIRETTA PER L'ITALIA PERSONAL COMPUTERS CON

ESCLUSIVO

## 4 ANNI DI GARANZIA\*



### TRE SOLUZIONI AI VOSTRI PROBLEMI:

#### AREA SERVICE

- ASSISTENZA TECNICA E MANUTENZIONE
- IN TUTTA ITALIA
- AUTOMATICA, ROBOTICA E TELEMISURE

#### AREA SOFTWARE

- SOFTWARE GESTIONALE E SCIENTIFICO
- STANDARD PERSONALIZZATO
- CORSI DI FORMAZIONE

#### AREA TRADE

- IMPORTAZIONE DI HARDWARE SPECIFICO
- RICERCHE DI MERCATO

**AREA SYSTEMS ITALIA s.r.l.** - 10137 Torino  
Corso Siracusa, 79 - Tel. (011) 3298580 - 351513 - Fax (011) 326872



COMPATIBILI AL 100% IBM\*

#### MP Plus CPU 8088/2

Clock 10/12 MHz 640 Ram

#### MP 286 CPU 808286

Clock 10/16 MHz espandibile  
fino a 4 Mb Ram in piastra madre

#### MP 386 CPU 80836

Clock 20/25 MHz 2Mb Ram on board

#### MP LCD PORTATILE

Video cristalli liquidi  
elettroluminescente e a plasma

nelle versioni:

8088 - 286 - 386

\* Regolare certificato di garanzia.

**A PARTIRE DA  
599.000 LIRE  
anche a L. 29.000  
mensili**

RICHIEDETECI MATERIALE ILLUSTRATIVO. SCONTO PER RIVENDITORI QUALIFICATI E QUANTITÀ