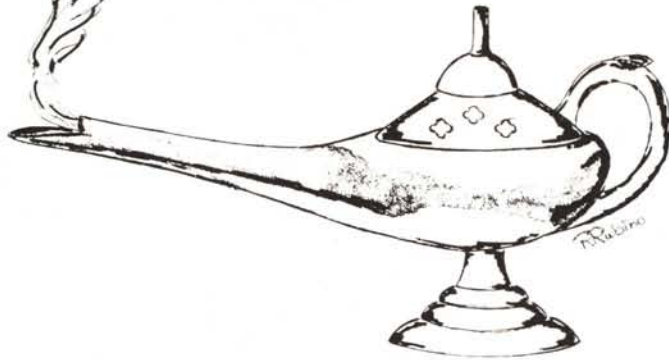




mister MSX



II PSG AY-3-8910

di Sergio e Dario Netti - Vicenza

Il PSG (Programmable Sound Generator) è un componente che ingloba, oltre a un generatore sonoro, anche 2 porte di I/O. Queste servono per controllare i joystick, paddle, touch-pad e l'ingresso seriale dati del registratore.

La sezione sonora dell'AY-3-8910 (prodotto dalla General Instruments, ma ne esistono equivalenti giapponesi) comprende 3 generatori di suono, un generatore di rumore, un mixer ed un generatore di inviluppo.

Il PSG ha 16 registri interni. Di questi, 14 sono i registri per programmare la sezione sonora e sono accessibili (in scrittura) da Basic tramite l'istruzione SOUND. Gli altri 2 sono in realtà le due porte di I/O.

Il PSG presenta sul bus di I/O tre registri (che non sono tre di quelli interni) che servono per avere accesso ai registri interni ed alle porte di I/O. Ecco i tre registri mappati come I/O :

&HA0: controllo; serve a selezionare uno dei 16 registri interni.

&HA1: Registro di scrittura.

&HA2: Registro di lettura.

Per scrivere un dato in uno dei registri si deve procedere così: con un'istruzione di I/O si scrive il numero del registro da selezionare all'indirizzo &HA0, poi si scrive il dato all'indirizzo &HA1. Analogamente si procede per leggere un registro, solo che la lettura viene effettuata all'indirizzo &HA2. Durante gli accessi al PSG è opportuno disabilitare l'interrupt per evitare errori di indirizzamento dei registri interni, in quanto pure la routine di interrupt può accedere al PSG.

Per rendersi conto di come funziona la cosa si possono eseguire delle prove in Basic, tenendo conto del fatto che a volte succedono degli errori per via dell'interrupt: ad esempio OUT &HA0, 8: OUT &HA1, 15 equivale a SOUND 8,15 ed il suo effetto è quello di mettere al valore massimo il volume del canale audio 0. OUT &HA0, 8: PRINT INP (&HA2) permette invece

di leggere il contenuto di tale registro. Notare che, anche se funziona a livello sperimentale, è bene non impiegare questo metodo per lavorare con il PSG anzi, è meglio non indirizzare direttamente l'hardware neanche in Assembly per non aver sorprese con qualche MSX «strano». Ribadiamo il concetto che è meglio servirsi del BIOS. A proposito di BIOS, quali sono le routine che servono a manipolare il PSG? Eccole:

&H0093: scrive nel registro (del PSG) puntato da A il dato contenuto nel registro E.

&H0096: legge nel registro A (dello Z80) il dato presente nel registro (del PSG) puntato da A.

I registri A ed E menzionati sopra sono quelli del microprocessore. L'impiego di queste routine da L/M è molto comodo, considerando anche che la routine di scrittura provvede da sola a disabilitare/abilitare l'interrupt nei punti critici. Ad esempio, per fare SOUND 13,8 da L/M potremo fare:

```
LD A,0DH
LD E,08H
CALL 0093H
RET
```

Le due ultime istruzioni potremo anche sostituirle con JP 0093H in quanto questa è la fine della mini-routine.

«Va bene — diranno i nostri 24 lettori (Manzoni ne aveva 25, ma non crediamo di essere alla sua altezza) — ma a che ci serve conoscere come accedere ai vari registri se non ne conosciamo il significato?».

OK, ecco la tabella tutta per voi:

R0 - R1: controllo frequenza canale A
R2 - R3: controllo frequenza canale B
R4 - R5: controllo frequenza canale C
R6: controllo frequenza rumore
R7: mixer e controllo porte I/O
R8: volume canale A
R9: volume canale B
R10: volume canale C
R11 - R12: controllo periodo inviluppo
R13: forma inviluppo
R14: port I/O
R15: port I/O

Ci rendiamo conto che queste cose sono di solito descritte nei manuali, ma spesso in modo poco chiaro. I primi 14 registri (da R0 ad R13) li conoscerete infatti tutti perchè accessibili da Basic tramite l'istruzione Sound, gli altri due sono invece un po' meno conosciuti perchè inaccessibili da Basic. Vediamo comunque i registri in dettaglio (questo è in parte un ripasso: vedere nel n. 42 di MC - giugno 1985 l'ottimo articolo di M. Bergami):

Il contenuto di R0 ed R1 è legato alla frequenza generata dal canale A del PSG con questa relazione (tratta dal manuale Spectravideo, ma modificata: infatti quella riportata nel manuale dava le frequenze dimezzate rispetto al

valore calcolato: un LA a 440 Hz risultava a 220 Hz, cioè un'ottava più in basso):

$$\text{VALORE REGISTRI} = 3579545 / (32 \cdot \text{FREQUENZA})$$

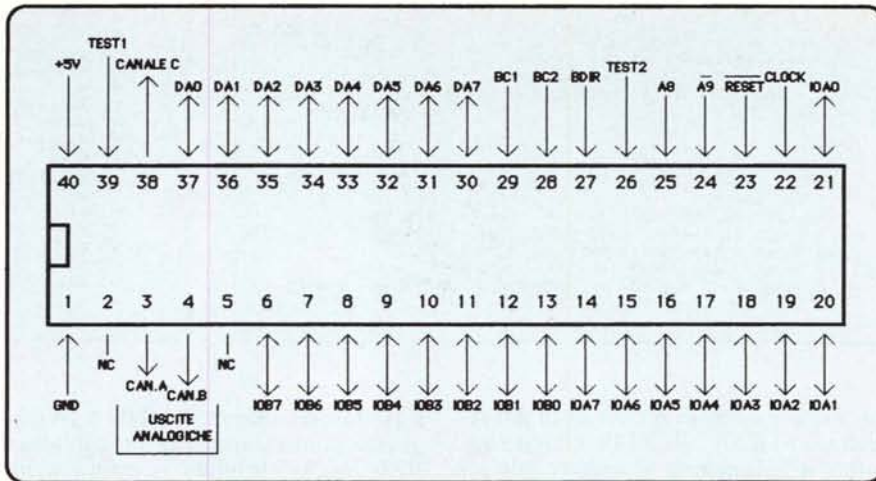
dove il numeraccio 3579545 rappresenta la frequenza del clock dell'MSX espressa in hertz. «Frequenza» rappresenta la frequenza che vogliamo ottenere in uscita da questo canale (espressa in hertz). Il «valore registri» che ci dà questa formula va posto (diviso in 2 byte, basso ed alto) appunto in R0 ed R1. Lo stesso discorso vale per R2 - R3 (canale B) ed R4 - R5 (canale C). Notare che i registri R1, R3 ed R5 sono a soli 4 bit, quindi se ci mettete dentro 0, 128 oppure 240 è come se ci metteste sempre 0, in quanto i 4 bit più significativi vengono ignorati. Questo si può verificare scrivendo un valore in un registro di questi: andandolo a rileggere i 4 bit più significativi sono sempre a 0. A questo punto possiamo calcolare il limite delle frequenze raggiungibili con il PSG: infatti il massimo numero binario ammesso per una coppia di questi registri è 0000111111111111, corrispondente al decimale 4095; dalla formula inversa:

$$\text{FREQUENZA} = 3579545 / (32 \cdot \text{VALORE REGISTRI})$$

Ci ricaviamo la frequenza: 27,316 Hz (frequenza minima). In questa formula c'è qualcosa di strano: infatti se poniamo il valore dei registri a 0 allora abbiamo una divisione per 0! Comunque se vengono posti a zero equivale ad 1, (cioè 0 in R1 ed 1 in R0) ed allora abbiamo la frequenza massima (inudibile) di 111860,7 Hz.

Adesso vediamo il registro R6: regola il periodo del generatore di rumore ed i bit utili sono 5, quindi i valori ammessi vanno da 0 (rumore più «acuto») a 31 (rumore più «cupo»).

Registro R7: corrisponde al mixer ed al controllo delle porte di I/O. I 3 bit meno significativi (0-2) servono ad abilitare il tono generato dai canali A (bit 0), B (bit 1) e C (bit 2). Ognuno di questi bit se posto a 0 abilita la propria funzione, se posto ad 1 la disabilita. I bit che vanno da 3 a 5 rappresentano una cosa analoga ma per il rumore: se il bit 3 è a 0 viene miscelato del rumore al canale A, se il bit 4 è a 0 accade lo stesso per il canale B ed il bit 5 pure per il canale C. I bit 6 e 7 non hanno significato in Basic per l'istruzione SOUND. Infatti questi bit controllano la direzione delle porte di I/O corrispondenti ai registri R14 ed R15. Il loro stato è ben definito e l'istruzione Sound non permette di alterarli. Ben diverso è il discorso lavorando in linguaggio macchina: infatti quando si scrive nel registro R7 da L/M bisogna tener presente che lo stato del bit 6 deve essere 0 (programma il port R14 co-



me input) e quello del bit 7 deve essere 1 (programma il port R15 come output). Questo per esigenze hardware, in quanto una programmazione diversa porterebbe ad un blocco delle periferiche associate al PSG.

R8, R9 ed R10 controllano il volume rispettivamente dei canali A, B e C. I valori permessi vanno da 0 (zero volume) a 15 (massimo volume); il va-

lore 16 passa il controllo del volume al generatore d'involuppo.

R11 ed R12 si occupano del periodo dell'onda prodotta dal generatore d'involuppo: in pratica è come se si trattasse di un unico registro a 16 bit, la parte bassa corrisponde ad R11 e la parte alta ad R12. La formula per calcolare il valore da assegnare a questi due registri è:

$$\text{VALORE REGISTRI} = \text{PERIODO} \cdot 3579545 / 512$$

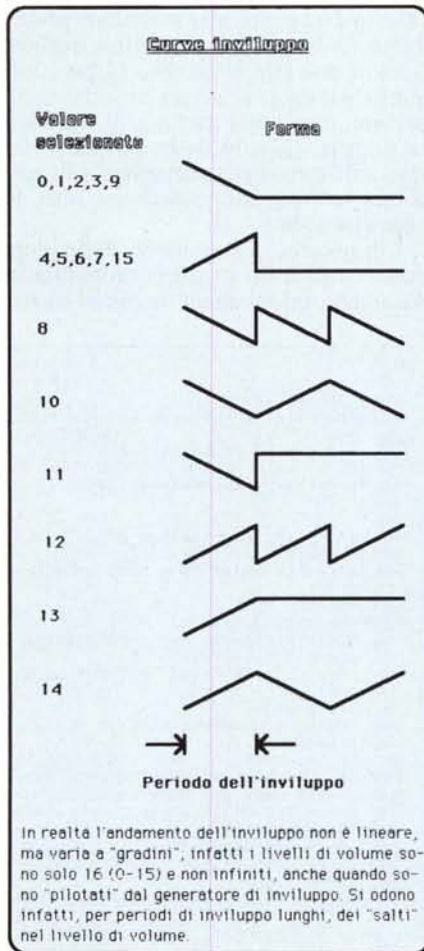
dove il periodo è espresso in secondi ed il risultato «valore registri» va inserito nei registri R11 ed R12, nella stessa maniera in cui si agisce per le prime coppie di registri del PSG.

Ponendo il valore di questa coppia di registri pari al massimo, cioè 65535, il periodo è di 9,37 secondi; con il valore pari ad 1 il periodo è di 0,14 millisecondi.

Vediamo ora l'ultimo registro sonoro: R13. Questo si occupa di selezionare il tipo di involuppo desiderato, cioè la sua forma d'onda. Vedere la figura relativa alle 8 possibili forme. Questa figura dovrebbe esserci in ogni manuale e credo chiarisca bene l'andamento del livello di volume nel tempo. Tener presente che l'involuppo «parte» all'atto della scrittura in R13. Alcune di queste forme sono periodiche, cioè si ripetono continuamente generando un suono continuo modulato in ampiezza: sono la 8, la 10 e la 14 (questi numeri rappresentano i valori da introdurre nel registro R13 per ottenere l'involuppo desiderato).

Con gli altri valori la curva d'involuppo «parte» quando si scrive in R13 e poi, esaurito il periodo d'involuppo, il volume rimane costante fino alla successiva scrittura nel registro.

Per i registri R14 ed R15 non abbiamo molto da dire: per chi programma in Basic non ha molta utilità il sapere che ci sono. Chi programma in L/M potrà al massimo divertirsi a far qual-



Listato 1

```

CALCOLO VALORI FREQUENZE
10 *****
20 *
30 * CALCULO VALORI FREQUENZE *
40 *
50 *****
60 *
70 INPUT "FREQUENZA";F
80 HL=3579545#/(32*F)
90 L=HL AND 255
100 H=INT((HL AND &HFD0)/255)
110 SOUND 0,L:PRINT "BYTE BASSO:";L
120 SOUND 1,H:PRINT "BYTE ALTO ":";H
130 SOUND 7,&B111000 * ABILITA I 3 CANALI SONORI
140 SOUND 8,15 * PORTA AL MASSIMO IL VOLUME DEL CANALE 0
150 GOTO 70

```

Listato 2

```

MUSICA SULL'INTERRUPT
10 *****
20 *
30 * MUSICA SULL'INTERRUPT *
40 *
50 *****
60 *
70 ON INTERVAL=18 GOSUB 100
80 RESTORE:SOUND 7,&B111000:INTERVAL ON
90 PRINT "PROVA "':GOTO 90
100 READ A,B:IF A=-1 THEN RESTORE:GOTO 100
110 SOUND 4,A:SOUND 5,B:SOUND 10,15:RETURN
120 DATA 190,0,170,0,214,0,172,1,29,1,29,1,0,0,29,1
130 DATA 172,1,214,0,170,0,190,0,190,0,0,0,-1,-1

```

cosa con il segnale proveniente dal registratore: il bit 7 di R14 è addetto appunto alla funzione di leggere tale segnale. Diciamo che non è molto utile agire direttamente su tali registri per leggere i joystick, paddle, ecc. in L/M perché il BIOS ci viene in aiuto con delle provvidenziali routine dedicate. Ecco gli indirizzi di queste routine:

&H00D5: Legge il joystick (o tasti cursore) puntato dal contenuto di A e ritorna il valore letto sempre in A.

&H00D8: Legge il trigger (o barra spaziatrice) puntato dal contenuto di A e ritorna il valore letto sempre in A.

&H00DB: Fornisce lo stato di un touchpad (mettere in A il parametro: 0-7); ritorno lettura nel registro A.

&H00DE: Fornisce lo stato di un paddle (mettere parametro nel registro A: 1-12); ritorno lettura nel registro A.

Il registro A menzionato sopra è quello della CPU ed i valori richiesti/forniti da queste routine equivalgono a quelli usuali del Basic (infatti il Basic utilizza proprio il BIOS per leggere queste periferiche).

Adesso che abbiamo finito con le descrizioni vediamo qualche esempio pratico di tutto quello che abbiamo esposto in merito al PSG: iniziamo dalla sezione sonora (e qui vi diamo un compito per casa: andate a consultare il già citato n. 42 di MC che spiega in modo forse più chiaro del nostro le varie funzioni sonore).

Esempi pratici

Se vogliamo ottenere un suono di frequenza 1000 Hz sul canale A potremo applicare la formula sopra esposta e ricavarci i valori da assegnare ad R0 ed R1: $3579545/(32*1000)$ dà come risultato 111, che, diviso in 2 byte dà proprio 111 da mettere in R0 e 0 da mettere in R1. Quindi faremo: SOUND 0,111: SOUND 1,0. Fatto questo non udremo ancora nulla in quanto dovremo abilitare il mixer ed il volume. Per attivare il canale A sul mixer basta fare: SOUND 7, &B00111110. Notare la forma binaria per mettere in risalto i vari bit. Per abilitare il volume dovremo agire su R8, quindi, volendo posizionare il volume

a 10 dovremo fare: SOUND 8,10 ed a questo punto udremo la nostra nota a 1000 Hz. Controllate il primo esempio: si tratta di un mini-programma di utilità che serve a determinare i valori da assegnare ad una coppia di registri in funzione della frequenza che si desidera ottenere. Se volete delle note musicali allora tenete ben presenti le regole musicali: la scala musicale si compone di 12 semitoni, corrispondenti alle note DO, DO#, RE, RE#, MI, FA, FA#, SOL, SOL#, LA, LA#, SI. Dopo il SI viene il DO dell'ottava superiore. Il LA 4 ha frequenza di 440 Hz e per calcolare la frequenza relativa al semitono successivo basta moltiplicare (o dividere per trovare la frequenza del semitono precedente) la frequenza per radice dodicesima di due (circa 1,0595). Dopo 12 di questi passaggi si arriva all'ottava superiore, composta da note di frequenza doppia. Quanto detto dovrebbe essere sufficiente per permettervi di calcolare le frequenze relative a tutte le note che volete.

Conoscere le frequenze delle note non è utile solo ai programmatori in Assembly: infatti anche restando in Ba-

sic può risultare necessaria una musicchetta che si ripete di continuo, da utilizzare come sottofondo per un gioco, ad esempio. Con l'istruzione PLAY si ottiene la musica, ma l'esecuzione di questa istruzione blocca per un certo tempo il programma. Ecco allora il trucco: si tratta di realizzare una routine agganciata all'interrupt tramite l'istruzione ON INTERVAL GOSUB. Questa routine deve leggere da alcune linee di DATA i valori per i registri sonori e li deve inviare al PSG tramite l'istruzione SOUND. In questo modo l'esecuzione della musicchetta non «intralca» il programma principale. Per realizzare una pausa si pone il valore per i registri sonori R4 ed R5 a zero. Questo, come spiegato precedentemente, crea una frequenza ultrasonica, quindi equivale ad una pausa. Per «allungare» la durata di una nota, invece, si provvede a ripeterla di seguito senza pause in mezzo.

Usò combinato di PLAY e SOUND

Forse non tutti sanno che l'istruzione SOUND può essere utilizzata abbinata all'istruzione PLAY con risultati

Listato 3

```

MINI SOUND EDITOR
10 *****
20 *
30 * MSX MINI SOUND EDITOR *
40 *
50 *****
60 *
70 *
80 KEY OFF:SCREEN 0:WIDTH 37
90 DIM SD$(13)
100 LOCATE 6:PRINT "MSX MINI SOUND EDITOR"
110 X=10
120 RESTORE
130 FOR Y=0 TO 13
140 SD$(Y)="000":IF Y=7 THEN SD$(Y)="184"
150 SOUND Y,VAL(SD$(Y))
160 LOCATE X,Y+4:PRINT "SOUND ";USING "## ";Y:PRINT SD$(Y)
170 NEXT
180 X=X+9:Y=0
190 LOCATE X,Y+4:PRINT SD$(Y):LOCATE X+4
200 IK$=INPUT$(1)
210 IF IK$=CHR$(30) THEN GOSUB 270:Y=Y-1:IF Y<0 THEN Y=13
220 IF IK$=CHR$(31) THEN GOSUB 270:Y=Y+1:IF Y>13 THEN Y=0
230 IF IK$=CHR$(13) THEN GOSUB 270
240 IF INSTR("0123456789",IK$)=0 THEN 190
250 SD$(Y)=MID$(SD$(Y),2)+IK$
260 GOSUB 290:GOTO 190
270 SD=VAL(SD$(Y)):IF SD>255 THEN LOCATE 9,22:PRINT "DATO NON VALIDO":RETURN 190
280 SOUND Y,SD
290 LOCATE 0,22:PRINT SPACE$(37):RETURN

```

interessanti. Volete una raffica di mitra? Provate:

PLAY«01T12058M640L4A»:SOUND6,15:-SOUND7,48

In questo esempio si sfrutta il macro linguaggio musicale per posizionare i parametri relativi alla durata ed alla forma dell'inviluppo. L'istruzione SOUND 6,15 posiziona il tono del generatore di rumore e l'istruzione SOUND 7,48 lo sovrappone alla nota generata dall'istruzione PLAY. Notare che se i vari parametri fossero stati dati tutti con delle istruzioni SOUND il suono sarebbe rimasto sempre attivo, infatti l'inviluppo 8 è periodico. L'istruzione PLAY provvede invece a zittire tutto alla fine della nota suonata.

Agendo in modo analogo a quello esposto sopra si può generare solo rumore con un'istruzione PLAY:

PLAY«T12050M3000L4A»:SOUND 6,18:SOUND7,55

In questo caso la nota A presente nell'istruzione PLAY è fittizia e serve solo a determinare la durata del suono.

Infatti viene «suonato» solo il rumore, modulato dall'inviluppo 0 selezionato dall'istruzione PLAY.

La sezione I/O in L/M

Vediamo brevemente come sfruttare le routine del BIOS associate al PSG e relative ai joystick (che sono tra le periferiche di input comuni: chi compra un MSX si procura infatti di sicuro un registratore ed un joystick come prima cosa).

```
XOR A ; Azzera A
CALL 00D5H ; Legge Stick 0
PUSH AF ; Salva
LD A,1
CALL 00D5H ; Legge Stick 1
POP BC ; Ripristina
OR B ; OR tra i valori letti
RET
```

Questa routine ritorna in A un valore (0-8) corrispondente alla posizione del joystick 0 (cioè tastiera) oppure joystick 1. Questo può essere utile per giochi L/M che prevedono l'input sia da tastiera che da joystick. In pratica equivale alla funzione STICK (0) OR STICK (1) del Basic, solo che il risultato risiede in A e deve essere utilizzato opportunamente dal programmatore per eseguire la funzione desiderata.

Analogamente si può agire per leggere i trigger:

```
XOR A ; Azzera A
```

```
CALL 00D8H ; Legge Strig 0
PUSH AF ; Salva
LD A,1
CALL 00D8H ; Legge Strig 1
POP BC ; Ripristina
OR B ; OR tra i valori letti
CALL NZ, BUM ; BUM se premuto
RET
```

La routine di indirizzo &H00D8 ritorna 0 se il tasto sottoposto al test non è premuto e 255 (cioè & HFF) se è premuto. Nel nostro esempio la routine BUM (che deve essere definita dall'utente) verrà eseguita pigiando lo spazio oppure il primo tasto di fuoco del joystick 1 (ricordiamo che lo standard prescrive 2 tasti, che non tutti i joystick hanno: sono presenti solo in quelli dichiarati MSX).

La lettura dello stato del secondo tasto dei joystick si può avere ponendo in A il valore 3 (joystick 1) e 4 (joystick 2) prima di effettuare la CALL 00D8H.

Per le altre routine relative a TOUCH-PAD e PADDLE non riportiamo esempi in quanto non abbiamo mai avuto a nostra disposizione queste periferiche e quindi non abbiamo potuto fare esperimenti in proposito, comunque riteniamo che quelle presentate siano le più interessanti.

CATANIA GIONS



Catania Gions, la prima avventura grafica per Apple realizzata in Italia. Una sfida alla tua abilità ed alla tua intelligenza, in mezzo ai pericoli ed alle insidie di un Egitto da cartolina, popolato da divertentissimi personaggi.

- Oltre 100 disegni
- Un vocabolario di 350 parole
- 200 azioni possibili
- Estrema rapidità di tracciamento delle immagini.
- Sist. Operativo custom

L.49.900 Iva comp.

Giotto, potentissimo editor grafico che, mediante una sofisticata tecnica di codifica, permette di racchiudere in un solo disco centinaia di disegni. Utilizzato nella realizzazione di Catania Gions.

- Interfaccia utente tipo Macintosh
- Sist. Op. ProDOS
- Routines per utilizzare i disegni all'interno dei vostri programmi.
- 768 colori disponibili
- 14 pennelli diversi
- Possibilità di disegnare a pieno schermo
- Animazioni e slide show

L. 80.000 Iva comp.



Catania Gions e Giotto sono disponibili su Apple][+ 64K, Apple //e, //c e IIGs (in emul.)

Entrambi i programmi possono essere richiesti a: **phonè**

Lungarno Gambacorti, 56 - 56100 PISA tel. 050/500136

Una realizzazione





QUEL CHE SI DICE AVERE LE SPALLE AL COPERTO

Il primo personal computer, l'Apple II, giunge in Italia nel 1979. Nel 1979 si costituisce la Automazione Sistemi Elettronici Microcomputers srf, con l'intento di sviluppare la propria presenza principalmente nel mercato dell'elettronica industriale. Aver scommesso nell'informatica già da subito, seguendo l'onda montante dei primi PET COMMODORE, APPLE II e dell'M20 OLIVETTI fa della ASEM, nel frattempo trasformata in S.p.A., una delle aziende più "vecchie" nel settore. Un'azienda a cui di "rumore" piace farne con le cose, con i risultati piuttosto che con le chiacchiere. Vi presentiamo pertanto la ASEM di oggi, risultato di indubbi successi di vendita, di capacità imprenditoriale e tecnica: un'azienda dal futuro sicuro. La progettazione della piastra elettronica, la realizzazione del ma-

ster e degli impianti da cui si ottiene il circuito stampato, la successiva saldatura dei componenti, le fasi di collaudo sono gli "steps" che il prodotto compie prima di essere immesso sul mercato o di essere avviato all'assemblaggio dei computers.

Nella scelta dei componenti si vincono e si perdono molte battaglie. Alla ASEM abbiamo salda una convinzione: il costo di un computer non è il prezzo che si paga: guasti, malfunzionamenti e inaffidabilità sono cose che devono essere messe in bilancio al momento dell'acquisto, come la serietà del produttore. È per questo che utilizziamo solo drives TOSHIBA, dischi fissi della NEC, tastiere CHERRY e PREH, alimentatori switching costruiti in conformità alle normative europee su nostre specifiche da una primaria azienda italiana.



Una fase del collaudo delle schede prima del montaggio dei computers. Sotto, una fase della realizzazione del master.



La forza vendite Asem è presente, in Italia, in quasi tutte le regioni: è attualmente allo studio la realizzazione di centri territoriali di assistenza tecnica.

Piemonte/Val d'Aosta	Mar Due Snc	011/3290769
Lombardia	PC Plus Srl	02/2841544/5
Veneto/Trentino A.A.	ECO Srl	045/916476
Friuli V.G.	Lucio Rodaro	0432/962282
Emilia Romagna	Marco Giannasi	0522/294805
Toscana/Umbria	G. Presentini e C.	055/973151
Lazio	Luigi Ricci	06/6237040
Campania/Calabria	S. Barbagallo	081/414994
Puglia/Basilicata	N.R. Cavallo	080/330499

 **ASEM**

Asem spa
Zona artigianale - Buia (UD)
telef. 0432/962282 telex 450608

COME USARCI AL MEGLIO

La espansione territoriale della ASEM in Italia e la costante crescita del numero di clienti e di fatturato ci ha imposto l'introduzione di alcuni aspetti organizzativi che devono essere conosciuti dai nostri clienti affinché essi possano trarre il massimo dei vantaggi da questi nuovi servizi.

Ordini

Per l'inoltro degli ordini, i signori rivenditori quando non vogliono contattare l'Agente di zona, possono trasmetterli direttamente all'Ufficio Vendite, che provvede inoltre all'invio della documentazione anche agli utenti finali avendo cura di indicare il punto vendita più vicino.

Hot Line

Una hot line facente capo all'Ufficio Tecnico è a disposizione per risolvere ogni problema di natura hardware. La hot line è attiva il pomeriggio e risponde al numero 0432/961014.

Assistenza

Sia le parti meccaniche sia le parti elettroniche sono soggette a guastarsi: è nell'ordine delle cose. Ciò che importa è avere le risorse tecniche ed umane in grado di intervenire in tempi accettabili; a tale fine va contattato l'Ufficio Assistenza.

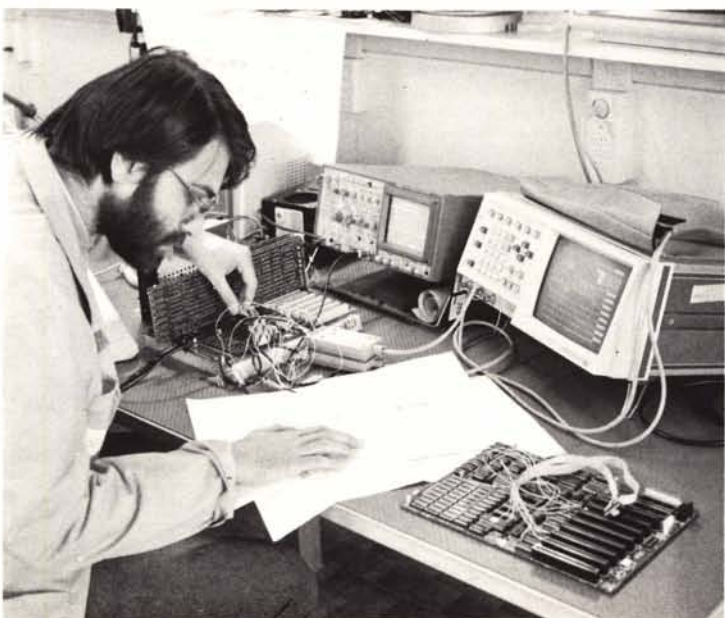
Spedizioni

ASEM si è sempre distinta per la celerità e la flessibilità nell'evasione degli ordini (di questa stessa opinione sono i clienti che ci seguono da anni); ciò non vuol dire che alle volte non serva sollecitare una spedizione o una riparazione che tarda.



Telefonateci o veniteci a trovare: imparerete a chiamarci per nome.

**ASEM
LO
STANDARD
E LA
DIFFERENZA**



THOMPSON È UNA MANNA PER GLI OCCHI

Cinque sono i modelli che compongono la gamma di monitor a colori della THOMPSON. Le caratteristiche comuni a tutti i modelli sono l'utilizzo di CRT di alta qualità antiriflesso e con sfondo nero per far meglio risaltare colori e caratteri.

Due sono i modelli di monitor a colori; il CM31311 SIR ed il CM31481 SIR, entrambi a 12". Si distinguono per la capacità risolutiva del tubo: il primo modello ha capacità pari a 0.31 mm DOT PITCH e 14 MHz di banda passante, mentre il secondo a 0.48 mm DOT PITCH e stessa banda passante. Entrambi vengono pilotati dall'adattatore grafico colore

del PC IBM. I due modelli a 14", CM36382 SIR e CM36512 VPIR, anch'essi si distinguono per la definizione del tubo, rispettivamente di 0.38 mm e di 0.51 mm DOT PITCH, con la stessa banda passante di 12 MHz. I dati relativi alla risoluzione sono analoghi ai 2 modelli a 12". Per la scheda ENHANCED GRAPHIC ADAPTER prodotta dalla IBM, THOMPSON ha progettato un modello specifico in grado di rispondere al meglio alle caratteristiche grafiche dell'adattatore. Riesce pertanto a visualizzare 650 H per 350 V punti con ben 64 colori possibili, disponendo di sincronismi automatici a 15.7 e a 22.0 KHz.