

In questa terza puntata parliamo delle routine in linguaggio macchina che servono per lo spostamento delle finestre, la traslazione dell'immagine nelle quattro direzioni, l'inversa ed il trasferimento da una all'altra delle pagine di alta risoluzione. Tutte le funzioni descritte nelle precedenti due puntate erano, con eccezione di una piccola routine per la lettura dei PADDLE, in BASIC, perché non era richiesta una particolare velocità. Nelle funzioni descritte questo mese, invece, trattandosi di manipolare in pratica tutti i 53760 punti dello schermo, un programma in BASIC usando i PEEK e POKE diventerebbe insopportabilmente lento e bisogna per forza ricorrere alla programmazione in linguaggio macchina ... NO! Non voltate subito pagina solo perché parliamo di linguaggio macchina. La programmazione in linguaggio macchina viene ritenuta da molti un'area riservata solo agli addetti ai lavori e perciò una cosa da evitare, difficilissima ed inavvicinabile. Non è vero! Non è più difficile o inavvicinabile del BASIC e del concetto di computer che all'inizio presentava non pochi ostacoli, ma che avete sicuramente superato. Inoltre, una buona conoscenza della programmazione in linguaggio macchina apre la strada ad un mondo di velocità di esecuzione dei programmi veramente emozionante. Insomma ASSEMBLER È BELLO!

Alcune delle routine presentate non hanno bisogno di passaggio di variabili dal programma in BASIC e perciò è sufficiente un semplice CALL per eseguirle. Così ad esempio la funzione INVERSE, che inverte il colore di tutti i punti dello schermo, viene eseguita con un CALL 6374 dal programma BASIC. Dal monitor può essere chiamata con 18E6G, che è l'equivalente di 6374 in esadecimale. Altre routine, come ad esempio MOVE WINDOW, hanno bisogno di alcune variabili usate nel programma in BASIC, (le coordinate degli angoli della finestra). Il modo più semplice di trasferire queste variabili è di assegnare un'area di memoria, normalmente nella pagina zero, e POKE'are in questa area i valori delle variabili dal BASIC:

POKE M,V
dove M è l'indirizzo della locazione di memoria e V è il valore della variabile (compresa tra 0 e 255, \$00 e \$FF in esadecimale). Poi dalla routine in linguaggio macchina possono essere riprese o manipolate per esempio con l'istruzione LDA \$M.

Il listato completo delle nuove routine è riportato nella figura 1 sotto forma di codice disassemblato, mentre la figura 2 riporta solo il codice macchina da inserire a partire dalla locazione di memoria \$17C5. Come di consueto, per inserire questi dati bisogna arrivare al monitor con il comando CALL-151. Poi si comincia ad inserire i dati partendo dalla locazione di memoria \$17C5:

* 17C5:85 19 0A 0A 29 1C 85 1B A5 19... etc.

Bisogna separare ciascun byte con uno spazio ed è opportuno battere RETURN



TAVOLETTA GRAFICA PER APPLE II

Terza parte

di Bo Arnklit

dopo ogni 7 o 8 righe, reimpostare l'indirizzo e continuare l'inserimento. Ad esempio dopo l'ottava riga battiamo RETURN e scriviamo:

* 1800:A9 20 8D D8 17 A5 ... etc.

Alla fine possiamo listare tutto scrivendo:

* 17C5.18FF <RETURN>

Alternativamente possiamo ottenere il listato della figura 1 impostando l'indirizzo \$17C5 seguito da L e RETURN, che fa apparire la prima "schermata" del listato. Premendo in successione L e RETURN si ottengono le rimanenti schermate. Come ulteriore controllo potete far girare il programma in Applesoft riportato nella figura 3 che calcola la somma di tutti i byte inseriti, che deve risultare 35270. È ovvio che questa somma può essere giusta senza che i dati lo siano: basta che ad esempio due dati siano scambiati tra di loro e la somma rimane costante. In ogni caso, comunque, se il risultato non è 35270 avete senz'altro fatto qualche errore. Fate particolare attenzione a distinguere tra la lettera B ed il numero 8, e tra la lettera D ed il numero 0. Infine la lettera O non esiste nella numerazione esadecimale e perciò tutti gli O sono per forza zeri.

Dopo aver inserito e controllato i dati bisogna salvarli sul disco con il nome ASS.CODE:

BSAVE ASS.CODE, AS17C0, LS83F
<RETURN>

Nel listato della figura 4 troverete le righe da aggiungere al programma della tavoletta grafica già pubblicato nei due numeri precedenti. Caricate il vecchio programma ed inserite tutte le righe della figura 4. È importante che non eseguiate una rinumerazione delle righe, altrimenti le istruzioni pubblicate in questa e nelle prossime puntate non andranno nei luoghi giusti.

Uso delle nuove routine

INVERSE

La funzione INVERSE serve per invertire il colore di tutti i punti dello schermo. In altre parole cambia tutti i punti bianchi in punti neri e viceversa, trasformando così l'immagine da positivo al negativo. Questa funzione può essere usata in qualsiasi momento e, ovviamente, se viene eseguita due volte di seguito l'immagine ritorna positiva. L'INVERSE è utilissimo non solo per ottenere l'effetto positivo/negativo ma anche quando si fa un errore nel disegno e si vuole cancellare la parte sbagliata disegnando sopra con il colore nero. Invece di cambiare il colore da bianco a nero si può eseguire la funzione INVERSE, cancellare con il bianco e ritornare al disegno originale.

le applicando ancora una volta la funzione INVERSE. Tra le funzioni in linguaggio macchina la INVERSE è la più semplice.

MOVE WINDOW

Nella puntata precedente avevamo pubblicato la funzione WINDOW che serve per definire un'area dello schermo. Ora, con le funzioni di MOVE WINDOW possiamo spostare (meglio replicare) questa finestra in qualsiasi parte dello schermo. Ci sono in tutto quattro funzioni relative al MOVE WINDOW. La funzione MOVE WINDOW OR sovrappone il disegno contenuto nella finestra al disegno che si trova nell'area dello schermo dove viene replica-

ta la finestra. La funzione MOVE WINDOW NORMAL invece replica direttamente la finestra cancellando così l'eventuale disegno presente nell'area destinata alla replica della finestra. L'applicazione della funzione MOVE WINDOW INVERSE è del tutto analoga ma, invece di replicare direttamente la finestra, tutti i byte vengono "complementati" per produrre una replica in negativo. Infine la funzione MOVE WINDOW X-OR ha l'effetto di sovrapporsi al disegno preesistente in maniera Exclusive Or. Vale a dire che un punto di colore uguale a quello del punto già presente sullo schermo nella posizione di plottaggio diventa nero mentre un pun-

to di colore diverso da quello già esistente viene plottato così come è. Quindi un disegno costituito da una linea bianca su fondo nero, spostato con la funzione MOVE WINDOW X-OR su un'area bianca, diventa una linea nera sempre sul fondo bianco. Per cancellare l'effetto del MOVE WINDOW X-OR basta eseguire la funzione un'altra volta per ripristinare perfettamente il disegno originale. Le funzioni MOVE WINDOW sono semplici da usare (bisogna solo prendere un po' di confidenza con i vari tipi). Dopo aver definito una finestra (vedi articolo su MCmicrocomputer numero 9) si sceglie una delle quattro funzioni dal menu. Muovendo i braccetti,

17C5-	85 19	STA	\$19	**** BASCAL ROUTINE ****	1865-	20 C5 17	JSR	\$17C5	
17C7-	0A	ASL		Calcola l'indirizzo di una riga	1868-	A0 00	LDY	\$00	
17C8-	0A	ASL		in HGR1 e HGR2	186A-	B1 00	LDA	(\$00),Y	trasferimento dei primi \$3F bytes della
17C9-	29 1C	AND	##1C		186C-	91 1A	STA	(\$1A),Y	riga da HGR2 a HGR1
17CB-	85 1B	STA	\$1B		186E-	C8	INY		
17CD-	A5 19	LDA	\$19		186F-	C4 3F	CPY	\$3F	
17CF-	6A	ROR			1871-	D0 F7	BNE	\$186A	
17D0-	6A	ROR			1873-	38	SEC		
17D1-	6A	ROR			1874-	A5 00	LDA	\$00	
17D2-	6A	ROR			1876-	E9 28	SEC	##28	
17D3-	29 03	AND	##03		1878-	85 00	STA	\$00	
17D5-	05 1B	ORA	\$1B		187A-	A5 01	LDA	\$01	
17D7-	09 40	ORA	##40	\$20=HGR1, \$40=HGR2	187C-	E9 00	SEC	##00	
17D9-	85 1B	STA	\$1B		187E-	85 01	STA	\$01	
17DB-	A5 19	LDA	\$19		1880-	C0 28	CPY	##28	
17DD-	6A	ROR			1882-	F0 09	BEQ	\$1880	
17DE-	29 E0	AND	##E0		1884-	B1 00	LDA	(\$00),Y	trasferimento dei rimanenti (40-\$3F)
17E0-	85 1A	STA	\$1A		1886-	91 1A	STA	(\$1A),Y	bytes della riga da HGR2 a HGR1
17E2-	6A	ROR			1888-	C8	INY		
17E3-	6A	ROR			1889-	C0 28	CPY	##28	
17E4-	29 18	AND	##18		188B-	D0 F7	BNE	\$1884	
17E6-	85 1A	ORA	\$1A		188D-	E6 3C	INC	\$3C	
17E8-	85 1A	STA	\$1A	indirizzo in \$1A, \$1B	188F-	E6 3D	INC	\$3D	
17EA-	60	RTS			1891-	CA	DEX		
17EB-	20 D7 18	JSR	\$18D7	**** ROUTINE H1->H2 ****	1892-	F0 10	BEQ	\$18A4	Prossima riga
17EE-	B1 3C	LDA	(\$3C),Y	Carica byte da HGR1	1894-	A5 3D	LDA	\$3D	
17F0-	91 3E	STA	(\$3E),Y	Memorizza byte in HGR2	1896-	C5 40	CMP	\$40	
17F2-	C8	INY		Loop 256 bytes	1898-	D0 AE	BNE	\$1898	Sequenza delle righe:
17F3-	D0 F9	BNE	\$17EE		189A-	A5 41	LDA	\$41	prima quelle da \$2D fino a 192
17F5-	E6 3D	INC	\$3D	Incrementa byte di indirizzo alto	189C-	85 40	STA	\$40	poi quelle da 0 fino a \$3D
17F7-	E6 3F	INC	\$3F		189E-	A9 00	LDA	##00	
17F9-	A5 3F	LDA	\$3F		18A0-	85 3D	STA	\$3D	
17FB-	C9 60	CMP	##60	continua fino a \$6000	18A2-	B0 A4	BCS	\$18A8	
17FD-	D0 EF	BNE	\$17EE		18A4-	60	RTS		
17FF-	60	RTS			18A6-	20 D7 18	JSR	\$18D7	**** ROUTINE SWAP ****
1800-	A9 28	LDA	##28	**** ROUTINE MOVE WINDOW ****	18A8-	B1 3C	LDA	(\$3C),Y	carica byte da HGR1
1802-	8D DB 17	STA	\$17DB		18AA-	AA	TAX		memorizza in Reg-X
1804-	A5 3D	LDA	\$3D	usa BASCAL per calcolare indirizzo	18AB-	B1 3E	LDA	(\$3E),Y	carica byte da HGR2
1807-	20 C5 17	JSR	\$17C5	della riga 40-191) contenuto in \$3D	18AD-	91 3C	STA	(\$3C),Y	memorizza in HGR1
180A-	18	CLC			18AF-	8A	TXA		trasferimento Reg-X in Accum
180B-	A5 1A	LDA	\$1A	Sommare all'indirizzo il valore in	18B0-	91 3E	STA	(\$3E),Y	memorizza in HGR2
180D-	65 41	ADC	\$41	\$41, \$42 e salva risultato in \$00, \$01	18B2-	C8	INY		LOOP per 256 valori di Y
180F-	85 00	STA	\$00		18B3-	D0 F3	BNE	\$18A8	
1811-	A5 1B	LDA	\$1B		18B5-	E6 3F	INC	\$3F	
1813-	65 42	ADC	\$42		18B7-	E6 3D	INC	\$3D	Incr. byte alto fino a \$4000
1815-	85 01	STA	\$01		18B9-	A5 3D	LDA	\$3D	
1817-	A9 40	LDA	##40		18BB-	C9 40	CMP	##40	
1819-	8D DB 17	STA	\$17DB		18BD-	D0 E9	BNE	\$18A8	
181C-	A5 3C	LDA	\$3C	Calcola indirizzo della riga	18BF-	60	RTS		
181E-	20 C5 17	JSR	\$17C5	contenuto in \$3C	18C0-	20 D7 18	JSR	\$18D7	**** Routine H2->H1 ****
1821-	A4 3E	LDY	\$3E		18C3-	B1 3E	LDA	(\$3E),Y	carica byte da HGR2
1823-	B1 1A	LDA	(\$1A),Y	Trasferire finestra da pag2 a pag1 con	18C5-	49 7F	EOR	##7F	invertire 1 bit
1825-	49 7F	EOR	##7F	larghezza \$3F-\$3E	18C7-	91 3C	STA	(\$3C),Y	memorizza in HGR1
1827-	91 00	STA	(\$00),Y		18C9-	C8	INY		
1829-	C8	INY			18CA-	D0 F7	BNE	\$18C3	loop 256 bytes
182A-	C4 3F	CPY	\$3F		18CC-	E6 3F	INC	\$3F	
182C-	D0 F5	BNE	\$1823		18CE-	E6 3D	INC	\$3D	incrementa byte alto fino a \$4000
182E-	E6 3C	INC	\$3C	Ripetere per righe successive fino ad	18D0-	A5 3D	LDA	\$3D	
1830-	E6 3D	INC	\$3D	arrivare al valore contenuto in \$40	18D2-	C9 40	CMP	##40	
1832-	A5 3C	LDA	\$3C		18D4-	D0 ED	BNE	\$18C3	
1834-	C5 40	CMP	\$40		18D6-	60	RTS		
1836-	D0 C8	BNE	\$1800		18D7-	A0 00	LDY	##00	Routine di inizializzazione puntatori
1838-	60	RTS			18D9-	84 3C	STY	\$3C	HGR1 e HGR2
1839-	18	CLC		**** MOVE GLOBAL ****	18DB-	84 3E	STY	\$3E	
183A-	A2 C0	LDX	##C0	carica Reg-X con 192 righe	18DD-	A9 20	LDA	##20	HGR1=\$2000 = \$3C, \$3D
183C-	A9 00	LDA	##00		18DF-	85 3D	STA	\$3D	HGR2=\$4000 = \$3E, \$3F
183E-	85 3C	STA	\$3C		18E1-	A9 40	LDA	##40	
1840-	A5 3D	LDA	\$3D		18E3-	85 3F	STA	\$3F	
1842-	85 41	STA	\$41		18E5-	60	RTS		
1844-	A9 C0	LDA	##C0		18E6-	A0 00	LDY	##00	**** ROUTINE INVERSE ****
1846-	85 40	STA	\$40		18E8-	84 0C	STY	\$0C	
1848-	A9 40	LDA	##40	calcola indirizzo della riga in \$3D (HGR2)	18EA-	A9 20	LDA	##20	\$0C, \$0D = \$2000 per HGR1
184A-	8D DB 17	STA	\$17DB		18EC-	85 00	STA	\$00	
184D-	A5 3D	LDA	\$3D		18EE-	B1 0C	LDA	(\$0C),Y	carica byte
184F-	20 C5 17	JSR	\$17C5		18F0-	49 7F	EOR	##7F	invertire bits
1852-	A5 1A	LDA	\$1A	somma all'indirizzo il valore in \$3E e	18F2-	91 0C	STA	(\$0C),Y	memorizza byte
1854-	65 3E	ADC	\$3E	memorizzare in \$00, \$01	18F4-	C8	INY		
1856-	85 00	STA	\$00		18F5-	D0 F7	BNE	\$18EE	loop 256 bytes
1858-	A5 1B	LDA	\$1B		18F7-	E6 00	INC	\$00	
185A-	69 00	ADC	##00		18F9-	A5 00	LDA	\$00	incr. \$0D fino a \$4000
185C-	85 01	STA	\$01		18FB-	C9 40	CMP	##40	
185E-	A9 28	LDA	##28		18FD-	D0 EF	BNE	\$18EE	
1860-	8D DB 17	STA	\$17DB	calcola indirizzo della riga in \$3C (HGR1)	18FF-	60	RTS		
1863-	A5 3C	LDA	\$3C						

Figura 1 - Listato della routine in linguaggio macchina.

17C5-	85	19	0A				
17C8-	0A	29	1C	85	1B	A5	19 6A
17D0-	6A	6A	6A	29	03	05	1B 09
17D8-	40	85	1B	A5	19	6A	29 E0
17E0-	85	1A	6A	6A	29	18	05 1A
17E8-	85	1A	60	20	D7	18	B1 3C
17F0-	91	3E	C8	D0	F9	E6	3D E6
17F8-	3F	A5	3F	C9	60	D0	EF 60
1800-	A9	20	8D	D8	17	A5	3D 20
1808-	C5	17	18	A5	1A	65	41 85
1810-	00	A5	1B	65	42	85	01 A9
1818-	40	8D	D8	17	A5	3C	20 C5
1820-	17	A4	3E	B1	1A	49	7F 91
1828-	00	C8	C4	3F	D0	F5	E6 3C
1830-	E6	3D	A5	3C	C5	40	D0 C8
1838-	60	18	A2	C0	A9	00	85 3C
1840-	A5	3D	85	41	A9	C0	85 40
1848-	A9	40	8D	D8	17	A5	3D 20
1850-	C5	17	A5	1A	65	3E	85 00
1858-	A5	1B	69	00	85	01	A9 20
1860-	8D	D8	17	A5	3C	20	C5 17
1868-	A0	00	B1	00	91	1A	C8 C4
1870-	3F	D0	F7	38	A5	00	E9 28
1878-	85	00	A5	01	E9	00	85 01
1880-	C0	28	F0	09	B1	00	91 1A
1888-	C8	C0	28	D0	F7	E6	3C E6
1890-	3D	CA	F0	10	A5	3D	C5 40
1898-	D0	AE	A5	41	85	40	A9 00
18A0-	85	3D	B0	A4	60	20	D7 18
18A8-	B1	3C	AA	B1	3E	91	3C 8A
18B0-	91	3E	C8	D0	F3	E6	3F E6
18B8-	3D	A5	3D	C9	40	D0	E9 60
18C0-	20	D7	18	B1	3E	49	7F 91
18C8-	3C	C8	D0	F7	E6	3F	E6 3D
18D0-	A5	3D	C9	40	D0	ED	60 A0
18D8-	00	84	3C	84	3E	A9	20 85
18E0-	3D	A9	40	85	3F	60	A0 00
18E8-	84	0C	A9	20	85	0D	B1 0C
18F0-	49	7F	91	0C	C8	D0	F7 E6
18F8-	0D	A5	0D	C9	40	D0	EF 60

Figura 2 - Listato del codice oggetto delle routine in linguaggio macchina.

le quattro crocette della finestra si sdoppiano e si spostano. Quando la finestra "mobile" è nella posizione desiderata si preme il pulsante ed istantaneamente il disegno viene replicato secondo la funzione scelta. Al momento della prima selezione di una delle funzioni MOVE WINDOW il disegno presente nella pagina 1 della grafica ad alta risoluzione viene duplicato nell'area di memoria relativa alla seconda pagina grafica. Poi durante l'esecuzione della routine di MOVE si usa quest'area come MASTER da trasferire "a pezzetti" nella prima pagina, in modo da avere sempre a disposizione l'immagine originale.

La funzione SHOW HGR2 serve appunto per visualizzare questa seconda pagina grafica e dopo uno spostamento della finestra si può vedere il disegno originale eseguendo la funzione SHOW HGR2. Se

si decide che tutto sommato il disegno era più bello prima del pasticcio combinato con le funzioni MOVE WINDOW si può usare la funzione SWAP o una delle funzioni H2->H1 per ripristinare la pagina 1.

La funzione SWAP effettua uno scambio delle due pagine grafiche. In pratica tutti i byte da \$2000 fino a \$3FFF vengono scambiati con quelli della pagina 2 che risiedono in memoria da \$4000 fino a \$5FFF.

H2->H1

Le funzioni H2->H1 servono per spostare la seconda pagina grafica nella prima pagina con quattro possibilità di trasferimento, come nel caso delle funzioni MOVE WINDOW. Insieme alla funzione H1->H2, che sposta semplicemente la pagina 1 alla pagina 2, sono utilissime quando si vuole fare dei disegni impegnativi. Ad esempio, dopo aver realizzato una parte di un disegno questa può essere "salvata" nella seconda pagina grafica con la funzione H1->H2. Poi si può procedere con il disegno e se a un certo punto si desidera il disegno originale basta riprenderlo con una delle funzioni H2->H1. Un'altra ap-

lavoro e si preme il pulsante. Ora spostando i braccetti si vede tutto il disegno "seguire" il movimento. Quando il disegno è nella posizione desiderata si preme di nuovo il pulsante e si torna in modo cursore con la crocetta lampeggiante, dopodiché si può scegliere un'altra funzione del menu oppure continuare con la funzione MOVE GLOBAL. È da notare che la funzione MOVE GLOBAL usa la seconda pagina grafica come buffer e perciò distrugge qualsiasi disegno presente lì. Quindi se avete un disegno importante salvato nella seconda pagina grafica bisogna portarlo nella prima pagina con la funzione SWAP e poi salvarlo su disco prima di usare la funzione MOVE GLOBAL.

Descrizione delle routine in linguaggio macchina

Le routine presentate questo mese si distinguono in due gruppi: quelle che operano su tutti i byte della pagina grafica (INVERSE, H1->H2, H2->H1) e quelle che operano solo su parte dei byte che costituiscono la pagina grafica (MOVE WINDOW e MOVE GLOBAL). La routine

```

10 FOR I = 6085 TO 6399
20 N = N + PEEK (I); NEXT
30 N = N - PEEK (6104) - PEEK (6181)
40 N = N - PEEK (6182) - PEEK (6341) - PEEK (6342)
50 PRINT "SOMMA DEI BYTES=" ;N
60 PRINT "RISULTATO GIUSTO=35270"

```

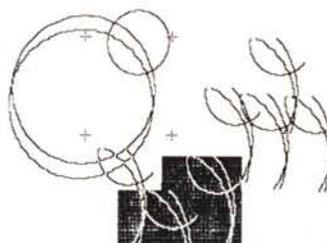
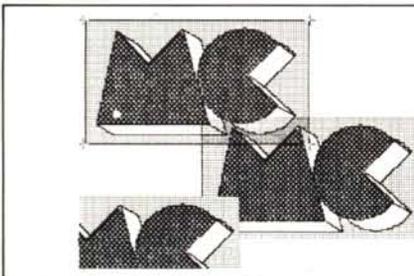
Figura 3 - Programmino di controllo correttezza dei dati immessi.

plicazione più creativa è quella di sovrapporre le due pagine grafiche con H2->H1 OR oppure H2->H1 X-OR. Consigliamo di esercitarsi un po' all'uso di queste funzioni (e dei MOVE WINDOW) per ottenere i risultati migliori.

MOVE GLOBAL

L'ultima funzione presentata questo mese è il MOVE GLOBAL che serve per lo spostamento dell'intera immagine nelle quattro direzioni. Questa è una funzione estremamente utile quando, ad esempio dopo aver fatto un disegno complicato, si scopre che magari sarebbe stato meglio se il disegno fosse un po' più giù e a destra. Con il MOVE GLOBAL si rimedia facilmente. Dopo averlo selezionato dal menu si portano i braccetti al centro del piano di

INVERSE è costituita da 26 byte e risiede in memoria a partire dalla locazione \$18E6. Le prime quattro istruzioni servono per creare un puntatore nelle locazioni \$0C e \$0D con i valori 00 e 20 rispettivamente. Questo è dunque l'indirizzo del primo byte della prima pagina grafica. Poi si entra nel loop da \$18EE a \$18F4 inizialmente con il registro Y uguale zero. Nel loop si usa l'indirizzamento indiretto indicizzato. L'istruzione LDA (#0C),Y significa: sommare il valore del registro Y al valore contenuto nelle locazioni \$0C e \$0D ed usarlo come indirizzo per la locazione di memoria da caricare nell'accumulatore. L'istruzione EOR #\$7F cambia tutti gli zeri in uno e viceversa e l'istruzione STA (\$0C),Y rimemorizza il contenuto dell'accumulatore allo stesso indirizzo di prima. Il loop viene effettuato 256 volte incremen-



Con le funzioni MOVE WINDOW è facile ripetere dei particolari di qualsiasi disegno.

```

1 REM *****
2 REM **
3 REM ** TAVOLETTA GRAFICA
4 REM **
5 REM ** COPYRIGHT 1982
6 REM **
7 REM ** BO ARNKLIT
8 REM **
9 REM *****
920 ON N% GOTO 1000,2000,3000,3000,3000,3000,4000,5000,6000,6000,6000,10000,11000,12000,13000,14000,
15000,16000,17000,18000
930 ON N% - 21 GOTO 19000,800,800,800,800,800,800,800,800,800,28000,29000,30000,31000,32000,33000,34000,35000
,36000,37000,38000,800
940 :
2000 REM MOVE GLOBAL
2010 GOSUB 40000:MN = 2:N% = 2: CALL 6123:GOSUB 90:MX = X%:MY = Y%
2020 GOSUB 40:GOSUB 270:XX = MX - X%:YY = MY - Y%:IF YY < 0 THEN YY = 191 + YY
2030 IF XX < 0 THEN XX = 279 + XX
2040 XX = INT (XX / 7)
2050 POKE 61,YY:POKE 62,XX:POKE 63,40 - XX:CALL 6201:GOSUB 210:IF PL = 0 THEN 2020
2060 PL = 0:PRINT B$:GOTO 2010
2070 :
3000 REM MOVE WINDOW
3010 IF WF = 0 THEN 810
3020 IF WW = 0 THEN GOSUB 500:CALL 6123:WW = 1:GOSUB 500
3030 POKE 6181,PL%(N% - 3):POKE 6182,PH%(N% - 3)
3040 MN = N%:GOSUB 40:IF Y% < 63 THEN 850
3050 GOSUB 40610:GOSUB 560:GOSUB 560:IF PL = 0 THEN 3040
3060 KX% = X% - (HW% - LW%) / 2:KY% = Y% - (HV% - LV%) / 2
3070 PL = 0:PRINT B$:GOSUB 500:GOSUB 40500:GOSUB 500:GOTO 3040
3080 :
4000 REM SWAP HGR1<->HGR2
4010 CALL 6309:GOTO 810
4020 :
5000 REM H1>H2
5010 CALL 6123:GOTO 810
5020 :
6000 REM H2>H1 OR,XOR,NORM,INV
6010 POKE 6341,PL%(N% - 5):POKE 6342,PH%(N% - 5):CALL 6336:GOTO 810
6020 :
10000 REM SHOW HGR2
10010 POKE H2,0:FOR I = 1 TO 50:GOSUB 210:NEXT :POKE H1,0:GOTO 810
10020 :
38000 REM INVERSE
38010 CALL 6374:GOTO 810
38020 :
40500 REM SUB MOVE W.
40510 W1 = LV%:IF KY% < 0 THEN W1 = W1 - KY%
40520 W2 = HV%:IF W2 - LV% + KY% > 191 THEN W2 = 191 + LV% - KY%
40530 W3 = LW%:IF KX% < 0 THEN W3 = W3 - KX%
40540 W4 = HW%:IF W4 - LW% + KX% > 279 THEN W4 = 279 + LW% - KX%
40550 POKE 60,W1:POKE 61,W1 + KY% - LV%:POKE 62,W3 / 7:POKE 63,W4 / 7:POKE 64,W2:POKE 65,ABS ((0
1 - W3) / 7):POKE 66,0
40560 IF O1 - W3 < 0 THEN POKE 65,256 - (W3 - O1) / 7:POKE 66,255
40570 CALL 6144:RETURN
40580 :
40600 REM MOVE WINDOW LIMITS
40610 X% = X% + 140:Y% = Y% - 65
40620 O1 = X% - (HW% - LW%) / 2:IF O1 < 0 THEN O1 = 0
40630 O2 = X% + (HW% - LW%) / 2:IF O2 > 279 THEN O2 = 279
40640 O4 = Y% + (HV% - LV%) / 2:IF O4 > 191 THEN O4 = 191
40650 O3 = Y% - (HV% - LV%) / 2:IF O3 < 0 THEN O3 = 0
40660 IF O1 > 278 THEN O1 = 278
40670 IF O3 > 190 THEN O3 = 190
40680 IF O2 < 1 THEN O2 = 1
40690 IF O4 < 1 THEN O4 = 1
40700 RETURN
40710 :
60060 PRINT D$:"BLOADASS.CODE,A$17C0"
60185 FOR I = 0 TO 7:READ PL%(I),PH%(I):NEXT
60260 DATA 17,0,81,0,234,234,73,127,17,60,81,60,234,234,73,127

```

Figura 4 - Righe di programma in Applesoft da aggiungere al programma della Tavoletta Grafica presentato nelle prime due puntate.

tando ogni volta il registro Y (INY). Alla fine di questo ciclo viene incrementata la locazione \$0D, cioè il byte più significativo dell'indirizzo, ed il ciclo ricomincia. Quando il valore di \$0D è uguale a 40 (esadecimale) si esce dalla routine e si ritorna al

programma Applesoft con l'istruzione RTS.

La routine H1->H2 è molto simile. Questa volta si creano due puntatori, uno per la prima pagina grafica ed uno per la seconda. Nel LOOP principale si carica un

byte dalla prima pagina e si memorizza nella stessa posizione nella seconda. La routine risiede tra \$17EB e \$17FF.

Le quattro routine di H2->H1 OR, X-OR, NORMAL e INVERSE sono in realtà una routine sola (\$18C0 - \$18D6) nella quale a seconda della funzione desiderata viene cambiata l'istruzione in memoria a \$18C5. Così per la funzione H2->H1 OR l'istruzione è ORA (\$3C), Y mentre per la funzione H2->H1 X-OR diventa EOR(\$3C), Y. La funzione H2->H1 NORMAL è ottenuta sostituendo l'istruzione con due NOP, cioè nessuna operazione, ed infine la funzione H2->H1 INVERSE usa l'istruzione EOR #\$7F. Il cambiamento di queste istruzioni viene eseguito dal programma in Applesoft (vedi riga 6010) che POKE'a direttamente in memoria i valori corrispondenti alle relative istruzioni.

La routine di SWAP è molto simile. Nel suo loop principale si carica l'accumulatore con un byte della pagina 1, e lo si trasferisce al registro X. Poi si carica un byte dalla pagina 2 e si memorizza nella pagina 1, ed infine si ritrasferisce il byte salvato in X all'accumulatore e si memorizza nella pagina 2. In questo modo si è effettuato lo scambio dei due byte usando come buffer il registro X. La routine risiede in memoria tra \$18A5 e \$18BF.

Come nel caso delle quattro funzioni H2->H1, le quattro funzioni di MOVE WINDOW sono implementate da un'unica routine in cui una istruzione viene modificata a seconda della funzione desiderata. La routine ha bisogno di 6 variabili, quattro delle quali definiscono le coordinate della finestra e due che determinano la distanza orizzontale e verticale tra la finestra fissa e quella mobile. Questi valori sono POKE'ati nelle locazioni \$3C - \$42 dal programma Applesoft. La routine usa la subroutine chiamata BASCAL (\$17C5 - \$17EA) per calcolare l'indirizzo di una delle 192 righe della pagina grafica. È necessario perché, purtroppo, nell'Apple II le righe grafiche non hanno l'indirizzo consecutivo ma seguono uno schema piuttosto complesso (vedi Reference Manual pag. 21).

La routine relativa alla funzione MOVE GLOBAL risiede in memoria dalla locazione \$1839 fino a \$18A4. È piuttosto lunga ma abbastanza semplice concettualmente. Dopo aver trasferito alla pagina 2 il contenuto della pagina 1 per avere una copia intatta del disegno, si tratta di riportarla alla pagina 1 cominciando però da un punto corrispondente allo spostamento orizzontale e verticale. Questi due valori sono trasferiti alla routine tramite i soliti POKE dal programma Applesoft (vedi riga 2050). In quest'ultima routine si nota in particolare l'elevatissima velocità di esecuzione che permette uno spostamento quasi in tempo reale.

Nel prossimo numero ci occuperemo delle funzioni di TEXT, che vi consentiranno di aggiungere scritte ai vostri disegni. **MC**

ACQUISTA LA TAVOLETTA GRAFICA!

Il prezzo è di L. 195.000

(compresa IVA, imballo e spedizione)

La tavoletta viene fornita completamente montata, calibrata e collaudata; è compreso il piano di lavoro con il menu su foglio di cartoncino plastificato e un dischetto Maxell MD1 con TUTTO il software, non solo quello presentato in queste pagine ma anche quello che seguirà nei prossimi numeri.

Il pagamento può essere effettuato tramite conto corrente postale n. 14414007 intestato a Technimedia s.r.l., via Valsolda 135, 00141 Roma o vaglia postale (in entrambi i casi compilate esattamente la causale del versamento e non inviate ulteriori comunicazioni postali). Per una maggiore rapidità, potete inviarmi una lettera con allegato assegno di c/c bancario o circolare intestato a Technimedia s.r.l. Infine, potete acquistarla direttamente presso i nostri uffici di Roma o in occasione di qualche mostra.