

software

SHARP PC-1500

Sottoprogrammi di gestione del video per il PC-1500

di Maurizio Sichera - Milano

SGV1500 è un sottoprogramma in linguaggio macchina che permette la visualizzazione di stringhe più lunghe di 26 caratteri e l'introduzione dei dati in modo più sofisticato di quello permesso dalla istruzione INPUT, e in modo più efficiente di quello permesso dalla funzione INKEYS.

SGV1500 occupa 497 byte di memoria a partire da un indirizzo qualunque: gli indirizzi che compaiono nel listato del programma sono puramente indicativi (vedi fig. 1).

Il sottoprogramma ha come argomento esplicito una variabile alfanumerica che contiene in ingresso la stringa da visualizzare e che viene, in genere, modificata in base ai caratteri digitati da tastiera. Come argomenti impliciti vengono usate le locazioni &77FD (lunghezza del prompt),

&77FE (parametro di controllo), &77FF (posizione del cursore) e la variabile ZS (terminatore).

Le operazioni svolte da SGV1500 possono essere più o meno complesse, secondo il valore del parametro di controllo. Il parametro stesso è codificato come segue:

&00 Il sottoprogramma visualizza i primi 26 caratteri della stringa, senza cursore.

&40 Il sottoprogramma visualizza la stringa - o una sua parte - con il cursore nella posizione specificata.

Figura 1

Codice oggetto del programma	0074:	77 F0 68 77	00F4:	8B 07 FD 60	0174:	A1 77 FF 91
SGV1500	0078:	6A FE 6D 02	00F8:	8B 07 8E 03	0178:	A2 2A FD 18
0000:	007C:	8B 24 87 08	00FC:	41 8B 03 A4	017C:	46 47 53 88
0004:	0080:	8B 82 87 0C	0100:	AE 77 FC 9A	0180:	04 A5 77 F0
0008:	0084:	8B 7E 6D 40	0104:	08 4A 0A 6D	0184:	B7 00 89 02
000C:	0088:	8B 18 6D 01	0108:	40 8B 40 60	0188:	B5 27 1E EF
0010:	008C:	8B 0C 87 18	010C:	25 4C 08 89	018C:	77 FC 01 F9
0014:	0090:	8B 1F 87 1C	0110:	0C DF A7 77	0190:	9A FD 5A FD
0018:	0094:	8B 2D 87 1D	0114:	FD 9B 40 91	0194:	58 9E 8B FB
001C:	0098:	8B 29 87 20	0118:	42 6F FF 8E	0198:	A1 77 FF 91
0020:	009C:	81 04 87 80	011C:	0D A7 77 FB	019C:	C6 2A A5 77
0024:	00A0:	81 21 FD 0A	0120:	93 4B DF A7	01A0:	FF DF FD DA
0028:	00A4:	48 7B 4A 80	0124:	77 FC 93 51	01A4:	FD 18 44 62
002C:	00A8:	A5 77 FC EB	0128:	6F 01 25 5A	01A8:	81 03 F5 88
0030:	00AC:	76 4E 01 FB	012C:	AF FD DA FD	01AC:	03 59 00 5C
0034:	00B0:	9A 62 67 61	0130:	98 FD 8B BE	01B0:	7B 89 02 5B
0038:	00B4:	60 60 DD 2E	0134:	E8 CA BE E4	01B4:	0D EF 77 FC
003C:	00B8:	B3 AF 1A B5	0138:	2C FD 0A FD	01B8:	FF F9 9A FD
0040:	00BC:	0D 51 5E 00	013C:	1A 86 99 3D	01BC:	5A FD 58 9E
0044:	00C0:	99 05 9A 2A	0140:	42 93 14 68	01C0:	E5 DD A7 77
0048:	00C4:	A5 77 FC 5A	0144:	77 6A FF 4A	01C4:	FF 91 F0 A5
004C:	00C8:	80 6E 1D 8B	0148:	01 9E 3F 6B	01C8:	77 FB A7 77
0050:	00CC:	CA 6E 1C 89	014C:	40 EB 78 80	01CC:	FF 91 F8 A5
0054:	00D0:	FE E9 77 F0	0150:	40 4C 08 89	01D0:	77 FF 2A DF
0058:	00D4:	00 8E 95 BE	0154:	01 62 62 25	01D4:	FD DA A5 77
005C:	00D8:	E6 69 F9 9A	0158:	DD 6A FF 2E	01D8:	F0 1E 24 A7
0060:	00DC:	ED 77 FF FF	015C:	DF A7 77 FB	01DC:	77 FC 8B 06
0064:	00E0:	9B 0B 44 44	0160:	81 02 6F FF	01E0:	81 04 EF 77
0068:	00E4:	FD 88 FD 18	0164:	9E 3A FD 5A	01E4:	FC 01 A7 77
006C:	00E8:	AE 77 FB DF	0168:	FD 58 9E 90	01E8:	FB 83 04 EF
0070:	00EC:	2A B5 00 28	016C:	A7 77 FB 9B	01EC:	77 FF 01 F9
	00F0:	AE 77 F1 F7	0170:	9A FD DA FB	01F0:	9A

&01 Il sottoprogramma visualizza i primi 26 caratteri della stringa, senza cursore, accetta un carattere da tastiera e ne restituisce il codice in Z\$.

&41 Il sottoprogramma visualizza la stringa - o una sua parte - con il cursore nella posizione specificata, accetta un carattere da tastiera e ne restituisce il codice in Z\$.

&02 Il sottoprogramma visualizza i primi 26 caratteri della &03 stringa, senza cursore, accetta un carattere da tastiera e ne esamina il codice: se questo è &0C (cursore a destra), posiziona il cursore sul primo carattere dopo il prompt; se il codice è &08 (cursore a sinistra), posiziona il cursore alla fine della stringa; in entrambi questi casi, somma &40 al parametro e continua ad accettare caratteri finché non viene introdotto un terminatore (vedi sotto). In tutti gli altri casi il sottoprogramma termina immediatamente e restituisce in Z\$ il codice del carattere introdotto.

&42 Il sottoprogramma visualizza la stringa - o una sua parte - con il cursore nella posizione specificata, poi accetta caratteri da tastiera e li memorizza in posizioni successive della stringa. Quando viene introdotto un carattere terminatore, il sottoprogramma ne restituisce il codice in Z\$ e termina. Sono considerati terminatori tutti i caratteri il cui codice è minore di &20 o maggiore di &7F, con l'eccezione di &08 e &0C (spostamento cursore), che vengono gestiti internamente.

&43 Come nel caso precedente, con la variante che anche i codici &18, &1C e &1D (CL, INS e DEL) vengono gestiti internamente e non sono considerati terminatori.

Per chiamare il sottoprogramma occorre quindi:

- a) Assegnare ad una variabile (GV negli esempi) l'indirizzo del sottoprogramma SGV1500, che coincide con il suo indirizzo di caricamento.
- b) Scegliere una variabile alfanumerica di lunghezza opportuna (B\$ (0) negli esempi) da usare come buffer di I/O ed assegnare a questa variabile il testo da visualizzare (eventualmente anche una stringa nulla).
- c) Decidere quanti caratteri della stringa costituiscono il prompt (e pertanto non sono modificabili) e memorizzare tale valore nella locazione &77FD; nel caso che questa informazione non sia significativa (parametro uguale a &00, &01, &40 o &41), memorizzare zero.
- d) Scegliere il valore del parametro e memorizzarlo nella locazione &77FE.
- e) Se il parametro ha valori maggiori o uguali a &40, memorizzare nella locazione &77FF l'indice del carattere su cui deve essere posizionato il cursore.
- f) Chiamare il sottoprogramma con l'istruzione CALL GV, B\$(0).

Al ritorno da SGV1500 si ha la seguente situazione:

- a) Il contenuto della variabile Z\$ è strut-

8	608	Cursore a sinistra	79	64F	O
9	609	Selezione Riserva	80	650	P
10	60A	Freccia in giu'	81	651	Q
11	60B	Freccia in su	82	652	R
12	60C	Cursore a destra	83	653	S
13	60D	ENTER	84	654	T
14	60E	BREAK (il programma BASIC viene interrotto alla fine dell'istruzione CALL)	85	655	U
15	60F	OFF	86	656	V
17	611	F1	87	657	W
18	612	F2	88	658	X
19	613	F3	89	659	Y
20	614	F4	90	65A	Z
21	615	F5	91	65B	Radice quadrata
22	616	F6	93	65D	Pi greco
24	618	CL	94	65E	Accento circonflesso
25	619	RCL	97	661	a
26	61A	CA (SHIFT/CL)	98	662	b
28	61C	INS (SHIFT/Cursore a destra)	99	663	c
29	61D	DEL (SHIFT/Cursore a sinistra)	100	664	d
30	61E	SHIFT/MODE	101	665	e
31	61F	MODE	102	666	f
32	620	SPACE	103	667	g
33	621	! (SHIFT/F1)	104	668	h
34	622	" (SHIFT/F2)	105	669	i
35	623	# (SHIFT/F3)	106	66A	j
36	624	\$ (SHIFT/F4)	107	66B	k
37	625	% (SHIFT/F5)	108	66C	l
38	626	& (SHIFT/F6)	109	66D	m
40	628	(110	66E	n
41	629)	111	66F	o
42	62A	*	112	670	p
43	62B	+	113	671	q
44	62C	,	114	672	r
45	62D	.	115	673	a
46	62E	-	116	674	t
47	62F	/	117	675	u
48	630	0	118	676	v
49	631	1	119	677	w
50	632	2	120	678	x
51	633	3	121	679	y
52	634	4	122	67A	z
53	635	5	128	680	DEF/SPACE
54	636	6	129	681	DEF/A
55	637	7	130	682	DEF/B
56	638	8	131	683	DEF/C
57	639	9	132	684	DEF/D
58	63A	:	133	685	DEF/E
59	63B	;	134	686	DEF/F
60	63C	<	135	687	DEF/G
61	63D	=	136	688	DEF/H
62	63E	>	137	689	DEF/I
63	63F	?	138	68A	DEF/J
64	640	@	139	68B	DEF/K
65	641	A	140	68C	DEF/L
66	642	B	141	68D	DEF/M
67	643	C	142	68E	DEF/N
68	644	D	143	68F	DEF/O
69	645	E	144	680	DEF/P
70	646	F	145	681	DEF/Q
71	647	G	146	682	DEF/R
72	648	H	147	683	DEF/S
73	649	I	148	684	DEF/T
74	64A	J	149	685	DEF/U
75	64B	K	150	686	DEF/V
76	64C	L	151	687	DEF/W
77	64D	M	152	688	DEF/X
78	64E	N	153	689	DEF/Y
			154	69A	DEF/Z
			157	69D	DEF/*

Figura 2 - Codici che possono essere restituiti al Basic.

to. Se il parametro valeva &00 o &40, il contenuto di Z\$ è casuale. In caso contrario, Z\$ contiene il codice del carattere che ha determinato l'uscita dal sottoprogramma. Il codice numerico del carattere può essere ottenuto come ASC (Z\$) o, in alternativa, come PEEK (&77F0).

b) Se il parametro valeva &02 o &03, il contenuto della locazione &77FE può essere stato cambiato in &42 o &43 rispettivamente.

c) Se la locazione &77FE al ritorno contiene &42 o &43, la variabile buffer B\$(0) è stata, in generale, modificata e la locazione &77FF contiene la posizione aggiornata del cursore.

d) La locazione &77FC contiene la lunghezza effettiva della stringa, mentre la locazione &77FB contiene la sua lunghezza massima.

Esaminiamo ora alcuni aspetti in maggior dettaglio.

SGV1500 accetta i caratteri da tastiera mediante la routine KEY SCAN II dell'interprete Basic (analogamente alla istruzione INPUT e diversamente dalla funzione INKEY\$, che usa la routine KEY SCAN I); quindi i tasti SHIFT, SML e DEF non generano un loro codice specifico, ma vengono usati come prefissi. I codici che possono essere restituiti al programma Basic sono tutti e soli quelli in figura 2.

All'ingresso nel sottoprogramma SGV1500 vengono fatti alcuni controlli di validità degli argomenti. Detta LP la lunghezza del prompt, LE la lunghezza effettiva

della stringa, LM la sua lunghezza massima e PC la posizione del cursore, deve essere:

- 1) LP <= LE
- 2) LP < LM
- 3) PC >= LP+1
- 4) PC <= LE+1
- 5) PC <= LM

(Le prime due condizioni vengono controllate sempre, le altre tre solo se deve essere visualizzato il cursore). Se i controlli non sono soddisfacenti, il sottoprogramma fa un Beep e ritorna immediatamente. In questo modo si evita che banali sviste nel programma Basic possano inchiodare la macchina o produrre altri effetti "strani".

Durante l'introduzione dei dati, la posizione del cursore viene tenuta sotto controllo, in modo da rispettare le condizioni sopra esposte. In particolare, l'introduzione di un carattere nella LM-esima posizione non fa avanzare il cursore; qualunque tentativo di spostare il cursore fuori dai limiti, o di usare INS quando la stringa contiene già LM caratteri, provoca un Beep di protesta e viene ignorato.

I tasti di spostamento del cursore ripetono il loro effetto automaticamente, se tenuti premuti abbastanza a lungo (questo effetto è ottenuto usando la routine KEY SCAN I dell'interprete).

Il tasto CL (se gestito internamente) cancella tutta la stringa, con l'eccezione dell'eventuale prompt.

Se nello spazio creato dal tasto INS non viene memorizzato un carattere, il simbolo di inserzione CHR\$(39) non viene eliminato e resta nella stringa che viene restituita al programma Basic.

Se i primi caratteri (max 6) della stringa sono numerici, è anche possibile avere uno stile di presentazione simile a quello che si ha durante la modifica di un programma Basic. Per ottenere questo effetto è sufficiente giocare sui valori del parametro: se ai valori elencati precedentemente viene sommata la costante &10, si ha una visualizzazione in cui il numero ed il resto della stringa, sono separati da uno spazio; se invece viene sommata la costante &14, il carattere separatore è ":". In linea di principio, l'uso di questa possibilità è completamente indipendente dalla presenza del prompt: tuttavia nella larga maggioranza dei casi i caratteri numerici iniziali saranno o tutti modificabili, o tutti protetti.

Se il programma Basic memorizza nella stringa le sequenze di due caratteri che rappresentano i "token" del linguaggio (ad esempio, CHR\$(&F1) + CHR\$(&92) corrisponde a GOTO), non vi sono problemi nella visualizzazione pura e semplice ed i token sono visualizzati espansi. Il sottoprogramma SGV1500 non è tuttavia capace di riconoscere i "token" durante lo spo-

```

1:"PROVA SGV1500
"
100:"A":CLEAR:CLS
:DIM A(20),B$(
0)*30
110:REM GU=256*
PEEK (&7865)+
PEEK (&7866)-4
97
111:INPUT "Base di
SGV1500: ";GU
200:"C":IF K<0LET
K=0
210:PA=3:IF K>20
LET K=20
220:P$="A("+STR$(
K)+")=":LP=LEN
(P$)
230:B$(0)=P$+STR$(
A(K))
240:POKE &77FD,LP
250:POKE &77FE,PA
260:CALL GU,B$(0):
Z=ASC(Z$)
270:IF PA>0LET Y=Z
:GOTO 290
280:IF ASC(INKEY$
)<>YLET PA=3:
GOTO 250
290:IF Y=10AND K<2
0LET K=K+1:PA=
0:GOTO 220
300:IF Y=11AND K>0
LET K=K-1:PA=0
:GOTO 220
310:IF Y<>13BEEP 1
:GOTO 260
320:IF PEEK (&77FE
)=3END
330:A(K)=VAL(MID$(
B$(0),LP+1,30
))
340:GOTO 230

```

Figura 3 - Programma Basic di esempio.

stamento del cursore e la cancellazione dei caratteri, per cui sul video si ottengono effetti decisamente "strani".

Un esempio

Come si può capire, SGV1500 si presta ad una varietà di applicazioni, anche sofisticate. A titolo di esempio riportiamo un programma Basic che usa SGV1500 in modo molto elementare, per esaminare e modificare gli elementi di un vettore (vedi fig. 3). Il programma ha bisogno di pochi commenti:

- 100 Dimensionamento del vettore da esaminare/modificare e della stringa da usare con SGV1500 (30 caratteri sono largamente sufficienti).

- 110-111 Se SGV1500 precede immediatamente il programma Basic, il suo indirizzo può essere calcolato automaticamente; in caso contrario, deve essere ottenuto con una istruzione INPUT (ovviamente si potrebbe usare anche READ/DATA).
- 200-210 Si controlla che l'indice della tabella sia dentro i limiti, in modo da evitare errori in esecuzione. Si assegna il valore 3 al parametro.
- 220 Costruzione del prompt.
- 230 Costruzione dell'intera stringa.
- 240 Definizione della lunghezza del prompt.
- 250-260 Chiamata di SGV1500.
- 270-300 Gestione dei tasti "freccia in giù" e "freccia in su", con ripetizione automatica.
- 310 Trattamento dei caratteri non gestiti.
- 320-340 Trattamento del carattere ENTER.

Sottoprogrammi ausiliari

In applicazioni complesse, il trattamento automatico dei caratteri può risultare insoddisfacente: occorre allora usare un parametro uguale a &40 o &41 e trattare i singoli caratteri nel programma Basic. Per facilitare questo trattamento, sono accessibili da programma Basic le tre funzioni fondamentali di trattamento caratteri contenute in SGV1500:

- a) memorizzazione di un carattere nella stringa;
- b) inserzione di un carattere nella stringa;
- c) cancellazione di un carattere della stringa.

Queste funzioni sono tutte ottenibili con normali istruzioni Basic, ma l'implementazione in linguaggio macchina risulta senza altro più efficiente. Tutti e tre i sottoprogrammi hanno come argomento esplicito la variabile alfanumerica su cui operare e trovano nella locazione &77FF l'indice della posizione su cui operare (se è invalido, si ha un Beep di protesta). I contenuti delle locazioni &77FD e &77FE vengono ignorati.

memorizzazione -- CALL base + 443, stringa

Il carattere da memorizzare viene preso da Z\$ (locazione &77F0); il contenuto della locazione &77FF viene incrementato, a meno che non sia già uguale alla lunghezza massima della stringa.

Inserzione -- CALL base + 358, stringa
Il carattere da inserire viene preso da Z\$ (locazione &77F0); tuttavia se Z\$="" (PEEK (&77F0)=0), si usa il simbolo di inserzione CHR\$(39). Il contenuto della locazione &77FF non viene modificato.

Cancellazione -- CALL base + 401, stringa
Il contenuto della locazione &77FF non viene modificato. **MC**

nuova

MICROLINE

La scrittura evoluta

Sono trascorsi oltre 5000 anni, la comunicazione si è evoluta attraverso i secoli per rispondere alle esigenze dei vari periodi.

NUOVA SERIE MICROLINE

Le nuove stampanti μ L sono la risposta alle esigenze dei nostri tempi: supersilenziose compatte, leggere, continuano la tradizione di affidabilità delle stampanti OKI.

μ L 182 80 colonne, 120 cps

μ L 192 80 colonne, 160 cps

μ L 193 136 colonne, 160 cps

bidirezionali ottimizzate, stampa qualità lettera, grafica alta risoluzione, set di caratteri 11 + IBM, inseritore automatico di fogli singoli (opzionale), menù di programmazione esterno, carrello regolabile, nastro a cartuccia autoinchiostro

... e tutte compatibili
PC IBM, APPLE MACINTOSH,
OLIVETTI, etc.



Il futuro presente

Dyneer

Technitron s.r.l.

Società del gruppo Dyneer Corporation
00199 ROMA - Via A. Vivaldi, 9
Tel. (06) 8395841-2 - 837035-6 - Telex: 680171 TECHRO I
20094 Assago (Mi) - V.le Milanofiori, Pal. E.2
Tel. (02) 8242112-3-4-5 - Telex: 341250 TECHMI I