

Alcune cose interessanti

Davide Casadio e Daniele Vasi - Ravenna

Sfogliando alcune riviste inglesi (vedi ad esempio YOUR COMPUTER n° 12 - 1983) siamo venuti a conoscenza di alcuni interessanti trucchetti e ritenendo che essi possano essere utili ai lettori di MC, in particolare agli utenti del COMMODORE VIC 20, proponiamo loro questo breve articolo con la speranza di vederlo presto pubblicato sulle pagine della vostra rivista.

Per proteggere i nostri programmi dai pirati di software esistono diversi metodi, più o meno efficaci.

Uno di questi consiste nel disabilitare la lettura dei tasti RUN/STOP e RESTORE e gli statement SAVE e LIST. Tutto ciò è possibile semplicemente modificando le locazioni di memoria che controllano queste funzioni.

La locazione 775 contiene normalmente il valore 199 e controlla l'esecuzione dello statement LIST. Se ne modifichiamo il contenuto (con un comando diretto oppure da programma) con POKE 775,198 otterremo che ogni tentativo di listare il programma porterà ad un blocco totale del sistema. Se invece digitiamo POKE 775,200 otterremo, in risposta ad un'eventuale richiesta di LIST, un messaggio d'errore.

Lo statement SAVE è controllato dalla locazione 818 contenente normalmente il valore 133, modificando il quale con POKE 818,32 si corrompe qualsiasi operazione di SAVE.

Il tasto RUN/STOP si disabilita con POKE 808,100 (la locazione 808 contiene normalmente il valore decimale 112). In tal modo, inoltre, ad ogni richiesta di LIST il computer rende indecifrabile il listato del programma.

Un metodo più breve ed altrettanto efficace è quello di digitare all'inizio del vostro programma la seguente linea:

```
10 POKE808,PEEK(808) + 2:POKE37150,PEEK(37150)AND127
```

In tal modo si disabilitano contemporaneamente i tasti RUN/STOP e RESTORE

VARIABILI DI SISTEMA DA 768 A 819	
BYTE	DESCRIZIONE
768-769	PUNTATORE AI MESSAGGI D'ERRORE
770-771	PUNTATORE ALLA ROUTINE DI GESTIONE DI UNA NUOVA LINEA BASIC
772-773	PUNTATORE ALLA ROUTINE DI CODIFICA DELLE PAROLE CHIAVE IN TOKENS
774-775	PUNTATORE ALLA ROUTINE D'IMPRESIONE DELLE PAROLE CHIAVE
776-777	PUNTATORE ALL'INIZIO DI UNA NUOVA LINEA BASIC
778-779	PUNTATORE ALLA ROUTINE PER L'ESECUZIONE DELLE OPERAZIONI ARITMET. ZONA IN CUI SONO CARICATI I VALORI DEI REGISTRI A,X,Y E P PRIMA DI UNA SYS
788-789	VEETTORE D'INTERRUPT HARDWARE (EABF)
790-791	VEETTORE D'INTERRUPT BRK (FED2)
792-793	VEETTORE D'INTERRUPT NMI (FEAD)
794-795	VEETTORE PER OPEN (F40A)
796-797	VEETTORE PER CLOSE (F34A)
798-799	VEETTORE PER ABILITARE L'INPUT (F2C7)
800-801	VEETTORE PER ABILITARE L'OUTPUT (F3B9)
802-803	VEETTORE PER RESET I/O (F3F3)
804-805	VEETTORE PER INPUT (F20E)
806-807	VEETTORE PER OUTPUT (F27A)
808-809	VEETTORE TEST DI STOP (F770)
810-811	VEETTORE PER GET (F1F5)
812-813	VEETTORE PER ABORT I/O (F3EF)
814-815	VEETTORE PER USR (FED2)
816-817	PUNTATORE PER LOAD (F549)
818-819	PUNTATORE PER SAVE (F685)

Figura 1

e gli statement LIST e SAVE. Per riportarsi alle condizioni iniziali digitare:

```
POKE808,PEEK(808)-2:POKE37150,PEEK(37150)OR127
```

Tuttavia questi sistemi sono efficaci solo se si fa eseguire il programma in cui sono contenuti. In caso contrario esso continuerà ad essere in balia di ogni pirata di software.

Il sistema che vi illustreremo ora vi permetterà di rendere invisibili ad occhi indiscreti le linee cruciali del programma.

La linea di programma che si desidera rendere "invisibile" deve contenere al massimo 17 caratteri, compresi il numero di linea e qualsiasi spazio. In fondo alla linea bisogna aggiungere:

```
:REM"
```

in tutto sei caratteri (non lasciate spazi!); ora premete RETURN e spostate indietro il cursore sino allo spazio immediatamente seguente il secondo apice. Premete poi DEL (una sola volta) in modo da cancellare il secondo apice. In seguito premete

SHIFT e INST esattamente 22 volte, poi rilasciate lo SHIFT e premete 22 volte DEL e premete RETURN.

Se ora listate il programma, la linea "trattata" apparirà per una frazione di secondo e subito sparirà senza lasciare traccia. Ma niente paura: la linea c'è ed il programma gira senza problemi.

Salvando su nastro e ricaricando il programma, si ottiene lo stesso effetto.

Le cose proposte dai nostri due amici sono interessanti e saranno senz'altro apprezzate dai lettori che sapranno farne buon uso. Cerchiamo però di ampliare qualche punto descrivendo il perché del comportamento diverso del sistema nelle situazioni sopra elencate.

Nelle prime quattro pagine della memoria del VIC sono contenute le variabili di sistema. I primi 1024 byte sono appunto usati per le variabili dell'interprete Basic (0-143); per le variabili del sistema operativo (144-255); stack (256-511); buffer del Basic (512-600); altre variabili del sistema operativo (601-767); indirizzi indiretti e vettori di salto (768-820); buffer di cassetta (828-1020).

La parte che ci interessa da vicino oggi è quella racchiusa tra gli indirizzi 768 ed 820 (daremo sempre valori decimali) che contengono i vettori di salto diretti e non diretti. Questa sezione come le altre dell'area in questione si trova in RAM e per questo assume una notevole importanza. Si possono infatti modificare gli indirizzi voluti, per modificare il modo di operare del sistema VIC 20, ottenendo di saltare routine del sistema operativo per indirizzare il sistema verso routine da noi introdotte.

In figura 1 riportiamo una tabella che potrà esservi utile per operazioni del genere. Nei byte 774 e 775 sono contenuti due numeri, 26 e 199, che puntano una certa locazione di memoria (precisamente $26 + 256 \times 199 = 50970$) e quando noi lanciamo il LIST il sistema legge il contenuto delle due locazioni sopra menzionate ed effettua il salto: se noi le modifichiamo, il salto avverrà in un'altra direzione.

Per farvi rendere conto praticamente di ciò, vi diremo che a partire dalla locazione di memoria 63636 inizia una routine che imprime sullo schermo il messaggio PRESS PLAY ON TAPE :

e se effettuerete una SYS 63636 lo vedrete comparire sullo schermo. Attenzione ora!

```
INT(63636)/256 = 248
```

```
248 x 256 = 63488
```

```
63636 - 63488 = 148
```

abbiamo con tali operazioni ricavato un puntatore a due byte che indirizza a 63636. Infatti una semplice verifica mostra che $148 + 256 \times 248 = 63636$

Provate ora a sostituire questo nuovo puntatore nelle locazioni 774 e 775 con POKE 774, 148 : POKE 775,248

e date il comando LIST (n.b. - in macchina deve esserci almeno una istruzione): vedrete comparire sullo schermo il messaggio in questione.

Lo stesso avviene per lo statement SA-

Elenco cassette per il VIC 20

Presso la redazione sono disponibili le cassette relative ad alcuni dei programmi pubblicati nella rubrica di software per il VIC 20. Per l'ordinazione inviare l'importo (a mezzo assegno, c/c o vaglia postale) a Technimedia srl, Via Valsolda 135, 00141 Roma.

codice	programma	MC n.	config.	lire
CVC/01	VIC-Maze	19	base	17000
CVC/02	Pic-man	23	base	17000
CVC/03	Briscola	25	base	17000
CVC/04	Grand Prix	28	base	17000
CVC/05	Frogger	26	6K	17000
CVC/06	Invaders	29	16K	23000
CVC/08	SKI	31	base	17000
CVC/07	Othello	29	16K	17.000

VE. Infatti modificando il contenuto delle locazioni 818 od 819 verrà modificato l'indirizzo di salto che non sarà più rivolto verso la routine di salvataggio programmi. Se sostituite in questi due byte rispettivamente i valori 34 e 253 che indirizzano alla locazione 64802 da cui inizia una routine di ripristino del sistema, dando il SAVE il sistema si bloccherà per 2-3 secondi dopo di che comparirà sullo schermo la scritta che viene inviata all'accensione della macchina.

Cose analoghe avvengono nelle altre situazioni. Si tenga presente che il sistema controlla in continuazione se è stato premuto il tasto STOP; inoltre il RESTORE si disabilita agendo sul registro di abilitazione dei flag d'interrupt di uno dei VIA interni al computer e mappato alla locazione 37150 (per maggiori chiarimenti si legga VIC DA ZERO di questo stesso numero).

Ski

di Gabriele Torresan - Treviso

Sono uno studente al 4° anno dell'Istituto Tecnico Commerciale indirizzo programmatori di Castelfranco Veneto ed appassionato lettore della vostra rivista che ho iniziato a comprare da più di un anno ed alla quale sono abbonato.

Sono in possesso di un VIC 20 e, come prima cosa, ho messo dentro tutti i pro-

Simboli "strani"

	= CTRL +8
	= CTRL +3
	= CTRL +7
	= CTRL +9
	= CTRL +0
	= CRSR VERT
	= CRSR ORIZZ.

grammi da voi pubblicati che girano su VIC inespanso.

Poi ho deciso di rimbocarmi le maniche e di creare qualcosa di mio. L'argomento del programma è lo sci. Il giocatore deve guidare uno sciatore che scende attraverso un bosco al termine del quale inizia una pista che va percorsa facendo lo slalom tra le bandierine (bisogna passare alla destra di quelle blu ed alla sinistra di quelle rosse) per giungere al traguardo dove ci attende una musicchetta ed il punteggio con il tempo impiegato.

Il programma gira perfettamente sul VIC in versione base ed è abbastanza veloce sfruttando la scroll-routine della macchina per simulare la discesa.

Il funzionamento del gioco è semplice: il calcolatore visualizza la prima pagina chiedendo all'utente di impostare la lunghezza della pista. Digitato un numero da 1 a 9, si inizia il gioco: lo sciatore parte da un box (in cui c'è anche un giudice di gara) ed inizia la discesa.

Premendo il tasto S l'omino vira a sinistra, premendo D vira a destra. Lo scopo è nel primo tratto della pista quello di evitare i pini, poiché essa attraversa un bosco. Alla fine del bosco inizia la pista vera e propria ai margini della quale ci sono dei cespugli che vanno sempre evitati. Durante lo slalom bisogna anche fare attenzione al

ghiaccio che ogni tanto si incontra sulla pista: passandoci sopra si può perdere il controllo per un attimo oppure scivolare e cadere. Nel malaugurato caso che lo sciatore vada a sbattere contro un pino, un cespuglio, una bandierina o scivoli sul ghiaccio, questi cade e sprofonda nella neve dalla quale spuntano fuori solo gli sci. A

```

1 REM *****
2 REM *
3 REM * S K I *
4 REM *
5 REM * G. TORRESAN *
6 REM *
7 REM *T. 0422/91240*
8 REM *
9 REM *****
10 PRINT "***** S K I *****":PRINT"BY GABRIELE TORRESAN"
20 PRINT"***** TASTI DA USARE *****":PRINT"0 SINISTRA"PRINT"1 DESTRA"
30 PRINT"***** LUNGHEZZA DELLA PISTA?*"
40 PRINT " [1-9]"
50 GETP$:IFP$<"1"ORP$>"9"THEN50
60 AZ=VAL(P$)*50
70 POKE36879,25:POKE36869,255:POKE650,128:POKE56,28:POKE52,28
80 PRINT"*****"
90 N=3
100 FORI=755207663:READR:POKEI,R:NEXT
110 K=7790:L=7612:H=8164:F=36884:G=38510:H=38532:X=11:M$="Z/ZZZZ/ZZZ"
120 FORI=0T0999:NEXT
130 PRINT"*****"
140 TI$="000000"
150 FORI=0TORZ
160 N=N+1
170 IFN=1THEN200
180 TFN=2THEN210
190 POKEM+8,56:POKEM+B+1,59:POKEF+B,5:POKEF+B+1,5
200 POKEM+A,55:POKEM+A+1,58:POKEF+A,5:POKEF+A+1,5:B=A
210 P=INT(RND(1)*21):IFPEEK(M+P)<32ORPEEK(M+P+1)<32THEN210
220 POKEM+P,54:POKEM+P+1,57:POKEF+P,5:POKEF+P+1,5:A=P
230 GOSUB610:NEXTI
240 N=0:0=2:S=0
250 FORI=0TORZ:GOSUB610
260 IFRND(1)<.1THEN0=INT(RND(1)*8)+8:POKEM+0,209:POKEF+0,3
270 Q=INT(RND(1)*4):POKEM+0,193:POKEF+0,5:0=INT(RND(1)*4)+18:POKEM+0,216:POKEF+0,5
280 N=N+1:IFN=8THEN0=0:0=INT(RND(1)*10)+6:POKEM+0,60:POKEF+0,4+0:0=-0
290 IFN<8THEN320
300 FORI=6T015:IFPEEK(L+I)=60*((0<0)*(<X>I)+<0>0)*(<X>I)THENS=S-ABS(X-I)*7:I=15
310 NEXTI
320 S=S+3:NEXTI
330 FORI=1T010:XX=X:GOSUB610:FORJ=0T089:NEXTJ,I
340 T$(0)="S"
350 T$(1)="T R A G U A R D O"
360 T$(2)="S"
370 FORI=0T02:PRINTT$(I):XX=X:GOSUB610:FORJ=0T089:NEXTJ,I
380 V=0:Z=0
390 FORI=1T05:PRINT"*****":XX=X:GOSUB670:FORJ=0T089:NEXTJ,I
400 FORI=1T06:XX=X:GOSUB670:FORJ=0T089:NEXTJ,I
410 POKEK+X,32:POKEK+X,32:POKEG+X,1:POKEH+X,1
420 FORI=1T06:PRINT:FORJ=0T089:NEXTJ,I
430 FORI=1T02
440 XX=X:GOSUB670:FORJ=0T089:NEXTJ,I
450 POKEK+X,61:POKEH+X,53
460 PRINT"*****UNTEGGIO:"S
470 PRINT"*****TEMPO:"
480 PRINT"*****MID$(TI$,4,1) MINUTI, "RIGHT$(TI$,2) SECONDI"
490 POKE36877,0:POKE36878,15
500 FORI=1T012:E=ASC(MID$(M$,I,1))+200
510 POKE36875,0:POKE36875,E:FORJ=0T0159:NEXTJ,I
520 POKE36878,0:POKE36875,0:GOTO570
530 POKE36878,15:POKE36877,180
540 FORI=15T00STEP-1:POKE36878,I:FORJ=1T020:NEXTJ,I
550 POKEK+X,150:POKEH+X,2
560 FORI=0T0999:NEXT
570 PRINT"*****IN ALTRA DISCESA(Y/N)?*"
580 GETP$:IFP$="Y"THENPOKE36869,240:POKE56,30:RUN
590 IFP$="N"THENPOKE36869,240:POKE36879,27:PRINT"*****":END
600 GOTO580
610 GETR$:Z=0:W=1:XX=X
620 IFR$="S"THEN660
630 IFR$="S"ANDX<0THENXX=XX-1:W=1:IFY<2THENZ=1:V=2:GOTO670
640 IFR$="D"ANDX<21THENX=XX+1:W=1:IFY<4THENZ=1:V=4:GOTO670
650 IFW=1THEN670
660 IFV=2ORV=4THENY=0:Z=1
670 POKEK+X,32:POKEK+X,32:POKEG+X,1:POKEH+X,1:PRINT:X=XX
680 IFPEEK(K+X)=209ORPEEK(L+X)=209THENX=X+(INT(RND(1)*7)-3):Z=1
690 IFPEEK(K+X)<32ORPEEK(L+X)<32THEN530
700 POKEK+X,V+48:POKEK+X,V+49
710 POKEG+X,4:POKEH+X,6:IFZ=0THEN740
720 POKE36878,0:POKE36877,250
730 FORJ=1T015:POKE36878,J:NEXTJ:FORJ=14T00STEP-1:POKE36878,J:NEXT
740 RETURN
750 DATA24,24,24,126,189,189,189,189,189,60,36,36,36,195,129
760 DATA56,48,48,124,186,186,186,186,186,44,41,58,44,200,176,32
770 DATA28,12,12,62,93,93,93,93,93,52,148,92,52,19,13,4
780 DATA0,1,0,1,3,0,1,3,7,1,3,7,15,1,3,7,15,31,1,3,7,15,31,63
790 DATA128,192,128,192,224,128,192,224,240,192,224,240,248,192,224,240,248,252,192
800 DATA224,224,248,252,254
810 DATA6,30,126,30,2,2,2,2
820 DATA77,77,77,127,28,28,28,28

```

GRAND PRIX

Chiediamo venia ai lettori ed all'autore del programma GRAND PRIX pubblicato nel n. 28 per un errore presente nel listato 1 a pagina 102. In questo riquadro ripubblichiamo quindi il programma incriminato nella sua versione corretta.

Chi ha provato ad introdurre in macchina il programma fornito, dopo il RUN avrà visto comparire sullo schermo il messaggio di errore DEVICE NOT PRESENT - ?ERROR IN 48 poiché in tale linea era stata sostituita (in teoria solo momentaneamente per usi redazionali) un'istruzione di caricamento da disco al posto delle istruzioni di autoloading/run e poi dimenticata.

Questa volta la linea 48 è esatta; con essa si leggono i DATA della linea 65 ed i numeri in questa contenuti vengono scritti nel buffer di tastiera; l'effetto di tale operazione è equivalente a quello che si otterrebbe scrivendo

```
PRINT CHR$(31) (return)
```

```
LOAD (return)
```

```
RUN (return)
```

L'unica differenza è che LOAD e RUN compariranno sullo schermo in forma abbreviata; CHR\$(31) pone queste scritte in blu. Se esse si vogliono rendere invisibili basta sostituire al numero 31 nella linea 65 il numero 5 che permette di scrivere il testo in bianco.

Molti lettori si sono accorti da soli del problema. Ricordiamo comunque che, a parte questo, non ci sono altri errori nel listato e che il programma "gira" perfettamente.

```

1 REM *****
2 REM *** GRAND PRIX VIC 20 ***
3 REM *** PARTE 1 CARICATORE ***
4 REM ** DANIELE VASI - RAVENNA **
5 REM *****
6
7
8
9
10 POKE36879,42:PRINT"Q":FORI=1TOS:PRINTTAB(21)"Q":NEXT
12 PRINT"Q"
14 PRINT"Q"
16 PRINT"Q"
18 PRINTTAB(3)"Q"
20 PRINT"Q"
22 PRINTTAB(5)"Q"
24 PRINTTAB(5)"Q"
26 PRINTTAB(7)"Q"
28 FORR=7620T07670:READA:POKER,A:NEXT:FORR=7416T07432:POKER,0:NEXT
30 POKE52,28:POKE51,0:POKE56,28:POKE55,0
32 FORI=7168T07263:READA:POKEI,A:NEXT
34 FORI=1T04000:NEXT
36 POKE36879,122
38 GOSUB66
40 PRINT"LA GARA E' COMPOSTA DA10 GIRI ED HAI 2 AUTO A DISPOSIZIONE."
42 PRINT"MERCA DI CONCLUDERLA EVITANDO GLI AVVERSARIE LE MACCHIE D'OLIO."
44 PRINT" PER STERZARE USA"
46 PRINT"BUONA FORTUNA !":FORI=1T09000:NEXT:GOSUB66:POKE198,7
48 FORT=631T0637:READA:POKET,A:NEXT
49 END
50 DATA162,230,189,253,30,157,19,31,189,253,150,157,19,151,202,224
52 DATA255,208,239,162,252,189,0,30,157,22,30,189,0,150,157,22,150
54 DATA202,224,255,208,239,162,0,169,32,157,0,30,232,228,23,208,248,96
56 DATA4,126,126,36,36,255,255,60,0,0,16,40,16,0,0,0,68,40,16,40,68,0,0,0
58 DATA36,66,72,68,42,64,0,170,84,162,73,34,84,34,172,24,62,127,255,255,63,30,0
60 DATA12,14,31,149,252,248,120,48,48,120,248,252,149,31,14,12,60,255,255,36
62 DATA36,126,126,24,96,102,254,231,231,254,102,96,6,102,127,231,231,127,102,6
64 DATA255,255,255,255,255,255,255,255
65 DATA31,76,207,13,82,213,13
66 PRINT"TAB(6)"

```

N.B.: il lancio di questo programma determina l'autocaricamento (con overlay) della seconda parte (pubblicata sul n. 28).

FROGGER

Nella nota sul Frogger pubblicata nel numero 29 era stato ommesso il listato 1, che qui riproduciamo. Ce ne scusiamo con i lettori: (magra) soddisfazione sia che così il gioco (pubblicato nel numero 26) gira anche con 8 o più K RAM.

```

1 REM *****
2 REM ***** F R O G G E R *****
3 REM ***** RIGHE DA AGGIUNGERE *****
4 REM ***** PER QUALSIASI *****
5 REM ***** CONFIGURAZIONE RAM *****
6 REM *****
7 IFPEEK(642)=32ANDPEEK(648)=30THENGOTO 10
8 PRINT"ATTENZIONE":PRINT"POKE 642,32:POKE648,30:SYS64818"
9 PRINT"RILEGGERE IL PROGRAMMA":STOP

```

questo punto si può digitare Y per continuare oppure N per smettere di giocare.

* * *

È un programma carino e divertente con un'idea di progetto semplice ma efficace. Nel riquadrino di pag. 145 riportiamo i soliti simboli "strani" usati nel programma.

Chi possiede un VIC espanso e non vuol prendersi la briga di sfilare le cartucce di espansione può digitare in modo diretto: POKE 641,0 : POKE 642,16 : POKE 643,0 : POKE 644,30 : POKE 648,30 ed effettuare alla fine SYS 58232 dopo aver premuto insieme i tasti RUN/STOP e RESTORE.

Si ottiene in questo modo di spostare lo START ed il TOP della memoria e la mappa dei caratteri allo stesso punto in cui si trovano in un VIC senza espansione.

Buon divertimento!

Commento al programma

Inizializzazioni e visualizzazioni generali

10 - 50 Stampa la prima pagina con richiesta difficoltà

60 - 70 Inizializzazioni

80 Disegno box di partenza

100 Allocations caratteri speciali

110 - 140 Inizializzazione variabili e pausa

Ciclo di percorso nel bosco

190 Disegno parte inferiore pini

200 Disegno parte centrale pini

210 - 220 Disegno parte superiore pini

240 Inizializzazioni

Ciclo di percorso su pista

260 Disegno ghiaccio

270 Disegno cespugli ai lati della pista

280 Disegno bandierine blu e rosse

290 - 310 Controllo sull'esatta esecuzione dello slalom ed eventuale diminuzione del punteggio

Parte finale del percorso

330 Stampa parte di percorso senza bandierine né cespugli

340 - 370 Stampa striscione del traguardo

380 - 390 Inizializzazioni e stampa delle aste del traguardo

400 Continuazione di stampa pista

410 Cancella sciatore (perché deve passare sotto al traguardo)

420 Continuazione di stampa pista

430 - 440 Continuazione di stampa pista

450 Stampa sciatore con le mani alzate

Termine del gioco

460 - 480 Visualizza punteggio e tempo

490 - 520 Suona "peana"

530 - 560 Rumore caduta e stampa sci a V

570 - 600 Richiesta ripetizione gioco

Subroutine di accettazione spostamento e controlli

610 - 640 Inizializzazione variabili per spostamento destra o sinistra

670 Cancella sciatore e inserisce linea a fondo schermo (causa scroll)

680 Spostamento anomalo causato dal ghiaccio

690 Controllo se la pista è libera

700 - 710 Disegna sciatore

720 - 730 Rumore di curva sulla neve

750 - 820 Dati per la creazione dei caratteri speciali



**MANNESMANN
TALLY**

le stampanti a stock presso



silverstar
componenti e periferiche

Sede: 20146 Milano - Via dei Gracchi, 20 - Tel. (02) 4996 (12 linee) - Telex 332187
40122 Bologna - Via del Porto, 30 - Tel. (051) 522231
00198 Roma - Via Paisiello, 30 - Tel. (06) 8448841 (5 linee) - Telex 610511
10139 Torino - P.za Adriano, 9 - Tel. (011) 443275/6 - 442321 - Telex 220181
35100 Padova - Via S. Sofia, 15 - Tel. (049) 22338

MT 600

- Velocità di stampa: 600 LPM
- Doppia tecnologia di stampa: Data processing e scrittura carattere std
- Doppia risoluzione grafica: 100 x 100 punti per pollice e 60 x 75 punti per pollice.

Accessori

- BAR CODES OCR - A o B
- Interfaccia seriale RS 232
- Interfaccia parallela standard

MT 440 L/D

- Velocità di stampa: 200/400 cps o 50/100 cps ad alta definizione
- Caratteri per linea: 132 caratteri a 10 cpi

Accessori

- Stampa: a 2/4 colori
- Interfaccia: Seriale 24 V / RS - 232 o parallela standard
- Alimentazione foglio singolo: Automatico per 250 fogli formato 210 x 297 mm (UNI A4)

MT 160/180

- Velocità di stampa: 160/200 cps (carattere std) e 40 cps (in stampa di qualità)
- Caratteri per linea: 80/132 a 10 cpi
- Matrice di stampa: 9 x 7
- Alimentazione foglio: a frizione o a trattore
- Interfaccia: Seriale integrale 24 V./RS 232 C e a 8 BIT parallela
- Disponibile con opzioni: alimentatore fogli singoli o introduttore automatico

Costo estremamente contenuto

MT 80

- 80 colonne, 80 cps, stampa bidirezionale
 - Grafica indirizzabile a singolo ago
 - Matrice di stampa: 9 x 8
 - Alimentazione foglio: a frizione e con trattore di spinta
 - Interfaccia: a 8 BIT parallela o 2 K buffer RS 232
- Bassissimo costo**

PIXY PLOTTER

- Numero delle penne: 3
- Dimensione foglio: UNI A4
- Velocità di esecuzione: 200 mm/sec
- Risoluzione: 0,1 mm
- Ripetibilità: 0,3 mm
- Interfaccia: RS 232 C o parallela
- Esecuzioni circolari, ad arco, a spirale
- Adattatore per curve.

Realmente economico

