

```

###      ### ### ##### #####
##### ##### ### ### ### ### ### ###
### ### ### ### ### ### ### ### ###
###      ### ### ### ##### ### design
###      ### ### ### ### ### ### ###
###      ### ### ### ### ### ### ###
###      ### ### ##### ### #####
    
```

BOLLETTINO DEL CLUB UTENTI MICRO DESIGN

GIUGNO 1982

Dobbiamo ringraziare tutti i nostri soci degli incoraggiamenti che abbiamo avuto in occasione del "censimento".

Abbiamo avuto anche molti preziosi consigli che terremo in grande considerazione per poter sempre migliorare la qualita' di quello che vi offriamo.

D' altra parte dobbiamo pero' tirare le orecchie a quei soci che hanno fatto programmi interessantissimi e modifiche hardware utilissime ma che non ce le hanno comunicate. Secondo noi lo spirito di collaborazione e' importantissimo per l'esistenza del club utenti MICRO design, infatti le applicazioni, sia di software che di hardware, sono talmente ampie da non poter assolutamente essere coperte da un singolo individuo. E' percio' necessaria una stretta collaborazione per poter raggiungere obiettivi piu' ambiziosi senza dover ripetere lavori che magari altri hanno gia' effettuato.

Pertanto vi invitiamo a sottoporci ogni vostro lavoro che pensiate possa essere utile ad altri soci o che possa stimolarli a realizzare soluzioni migliori.

A tal proposito desideriamo ringraziare il sig. Fabrizio Aversa di Roma che ci ha inviato una serie di programmi per utilizzare il grafico con il BASIC 3.

Prendendo lo spunto da tali programmi un nostro collaboratore ha realizzato una routine che permette di accendere e spegnere i singoli punti del grafico sia sulla vecchia scheda 32 x 16 che sulla nuova scheda 80 x 24.

Questo e' un ottimo esempio di come la collaborazione dei nostri lettori puo' portare alla realizzazione di programmi veramente utili e interessanti.

Aspettiamo quindi le vostre lettere con programmi, progetti e quanto altro possa riguardare il nostro microcomputer, pubblicheremo i piu' interessanti e ricompenseremo gli autori dei lavori.

Se invece avete realizzato qualcosa che ritenete commercialmente valido inviatecene una breve descrizione, in seguito valuteremo insieme l'opportunita' e le condizioni di presentazione sul bollettino.

SUBROUTINE DI GESTIONE DEL PUNTATORE VIDEO.

In molti casi puo' essere comodo posizionare il cursore (che nella scheda interfaccia video 32 x 16 e' rappresentato da un quadratino mentre nella nuova scheda 80 x 24 e' lampeggiante) in zone diverse dello schermo, ad esempio facendolo ritornare verso l'alto senza provocare lo "scroll" cioe' lo spostamento di tutte le linee del video verso l'alto.

A tal proposito i nostri tecnici usano, da tempo, una utile subroutine scritta in BASIC 3 che consente di posizionare il cursore in una qualsiasi locazione del video indicando semplicemente la linea e la colonna in cui si desidera spostare il cursore.

La subroutine deve essere chiamata ponendo prima nelle variabili HT e VT rispettivamente i valori della colonna e della riga che si vuole occupare con il cursore.

Occorre tenere presente che le colonne vengono contate a partire da sinistra, mentre le righe vengono contate a partire dall'alto. Cioe' per posizionare il cursore nell'angolo in alto a sinistra del video occorrera' assegnare:

HT=1:VT=1

e quindi chiamare

GOSUB 60000

Di seguito riportiamo le liste della subroutine che potrete inserire nei vostri programmi, la prima si riferisce alla video da 32 x 16, mentre la seconda deve essere utilizzata da coloro che possiedono la nostra interfaccia programmabile da 80 x 24 caratteri.

```
60000 PU=PEEK(&48)+PEEK(&49)*2^8-&EC00:POKE PU+&EC00,&20
60010 VL=INT(PU/32):OL=INT(PU-VL*32)
60020 IF HT>0 THEN OL=(HT-1)MOD 32
60030 IF VT>0 THEN VL=(VT-1)MOD 16
60040 ZZ$=HEX$(VL*32+OL+&EC00)
60050 POKE VAL("&"+ZZ$),&81
60060 POKE &48,VAL("&"+RIGHT$(ZZ$,2)):
        POKE &49,VAL("&"+LEFT$(ZZ$,2))
60070 RETURN
```

```
60000 PU=PEEK(&48)+PEEK(&49)*2^8-&E800
60010 VL=INT(PU/80):OL=INT(PU-VL*80)
60020 IF HT>0 THEN OL=(HT-1)MOD 80
60030 IF VT>0 THEN VL=(VT-1)MOD 24
60040 ZZ$=HEX$(VL*80+OL+&E800)
60050 OUT &E8,15:OUT &E9,VAL("&"+RIGHT$(ZZ$,2)):
        OUT &E8,14:OUT &E9,VAL("&"+LEFT$(ZZ$,2))-&E8
60060 POKE &48,VAL("&"+RIGHT$(ZZ$,2)):
        POKE &49,VAL("&"+LEFT$(ZZ$,2))
60070 RETURN
```

USO DEL SEMIGRAFICO CON IL BASIC 3.

Il BASIC 3 e' realizzato per essere utilizzato su computer CP/M compatibili e, come ha potuto sperimentare chi lo possiede, e' uno dei piu' potenti e versatili disponibili sul mercato.

Pero' molti soci avevano scritto programmi che utilizzavano le istruzioni SET e RESET (ACCENDI e SPEGNI) del famoso Basic 1.0 italiana, inglese sviluppato nei nostri laboratori, pertanto abbiamo preso lo spunto da uno dei nostri soci: il signor Fabrizio Aversa di Roma che ci ha gentilmente inviato una serie di programmi matematici che utilizzano il semigrafico per disegnare grafici di funzione.

Pur essendo tali programmi interessantissimi per chi si occupa di elaborazioni matematiche molto complesse, abbiamo preferito riscrivere in modo piu' generale la subroutine che gestisce il semigrafico e farla leggere automaticamente all'inizio del programma che la deve utilizzare.

Vi presentiamo tale programma che servira' anche da esempio di come sia possibile utilizzare una subroutine in linguaggio assembler richiamata da BASIC 3.

La prima cosa da fare, se si vuole utilizzare una subroutine in linguaggio assembler da BASIC 3 consiste nello scriverla utilizzando il linguaggio assembler che si ha a disposizione. Nel nostro caso abbiamo utilizzato l'M80 della Microsoft per sfruttare pienamente le istruzioni Z80. Infatti in tale programma non ci sono salti assoluti ed e' pertanto possibile posizionare la subroutine in qualsiasi area di memoria.

Chi non possiede l'M80 puo' battere il programma in linguaggio macchina con il DDT facendolo partire dall'indirizzo 100h e salvandolo poi con l'istruzione:

```
SAVE 1 SET.COM
```

Quando si ha il file SET.COM su disco si puo' scrivere il programma BASIC che si preoccupera' di caricarlo da disco in memoria. A tal proposito vi ricordiamo che occorre riservare una parte della memoria per contenere il programma SET e cio' puo' essere fatto rispondendo alla domanda che il Basic pone all'inizio: Highest Memory? anziche' con un return con un indirizzo.

Il modo piu' semplice consiste nel tener presente la tabella che vi forniamo togliendo al valore indicato la dimensione del vostro programma.

CP/M da		Highest memory?	&5C00	- <dimensioni>
	28K		&5C00	
	32K		&6C00	
	36K		&7C00	
	40K		&8C00	
	44K		&9C00	
	48K		&AC00	
	52K		&BC00	
	56K		&CC00	

Ad esempio, avendo un CP/M da 48K e una subroutine lunga 79H come la nostra, occorre rispondere almeno AC00 - 79 (i numeri sono in esadecimale). Per sicurezza e' meglio riservare un po' di spazio in piu' in modo da averne a disposizione per eventuali modifiche.

Vi consigliamo quindi, per facilitarvi i conti, di rispondere &AB00 dove & significa esadecimale e AB00 deriva dalla sottrazione AC00-100. (abbiamo riservato 100 anziche' 79 per sicurezza.)

ROUTINE IN BASIC PER CARICARE IN MEMORIA IL SET.COM

Anzitutto occorre assegnare alla variabile FC il valore corrispondente all'indirizzo di memoria che abbiamo riservato (vedi pag. precedente). Quindi all'inizio del programma scriveremo:

```
10 FC=&AB00
```

e, di seguito la chiamata alla subroutine

```
20 GOSUB 55000
```

da questo punto in avanti potremo scrivere il nostro programma che utilizzerà le routine grafiche. In fondo aggiungeremo naturalmente la subroutine:

```
55000 OPEN #5,"I","SET.COM"
55010 FOR I=0 TO &79
55020 POKE FC+I,BYTE(#5)
55030 NEXT I
55040 CLOSE #5
55050 RETURN
```

notate che dopo aver eseguito la subroutine avremo in memoria il contenuto del file SET.COM a partire dall'indirizzo AB00.

Pertanto nel nostro programma (cioè dalla riga 21 in avanti) potremo chiamare la routine grafica con l'istruzione:

```
CALL FC,1,SX,SY
```

oppure

```
CALL FC,0,SX,SY
```

ove la variabile FC è già stata assegnata alla riga 10, il successivo parametro deve essere 1 per accendere oppure 0 per spegnere il punto, SX e SY sono le coordinate del punto, tenendo conto che, per comodità, abbiamo fissato l'origine sullo schermo in basso a sinistra punto (0,0).

Vi diamo ora la lista della routine assembler che esegue l'accensione o lo spegnimento del punto e che, come vi abbiamo già spiegato, dovrete registrare su disco sotto il nome SET.COM mediante il DDT a partire dall'indirizzo 100H.

```
0100 D1 7B D6 48 30 FC C6 48 16 FF 14 D6 03 30 FB C6
0110 03 14 5F E1 7D D6 A0 30 FC C6 A0 26 FF 24 D6 02
0120 30 FB C6 02 6F AF E5 7A 21 80 EF 01 50 00 ED 42
0130 3D 20 FB E3 4C E3 09 E3 AF BD E1 06 00 28 12 83
0140 20 04 CB C0 18 1B 3D 20 04 CB D0 18 14 CB E0 18
0150 10 83 20 04 CB C8 18 09 3D 20 04 CB D8 18 02 CB
0160 E8 7E 87 30 03 87 30 03 3E 80 77 7E 4F AF D1 BB
0170 7B 28 03 B1 18 02 2F A1 77 C9 00 00 00 00 00
```

```

.z80
;
; per video 80x24 o 32x16
;
0048 GLIN equ 48h ; 30h
00A0 GCOL equ 0a0h ; 40h
0050 COL equ 50h ; 20h
EF80 VEND equ 0ef80h ; 0ee00h
0000' aseg
org 0100h
; Subroutine per BASIC 3 con video 80x24
;
; LINEA schermo nel reg.D
; linea carattere (0,1,2) nel reg.E
0100 D1 start: pop de
0101 7B ld a,e
0102 D6 48 fc: sub GLIN
0104 30 FC jr nc,fc
0106 C6 48 add a,GLIN
0108 16 FF ld d,0ffh
010A 14 fb: inc d
010B D6 03 sub 3
010D 30 FB jr nc,fb
010F C6 03 add a,3
0111 14 inc d
0112 5F ld e,a
; COLONNA schermo nel reg. H
; colonna carattere (0,1) reg. L
0113 E1 a3: pop hl
0114 7D ld a,l
0115 D6 A0 fc1: sub GCOL
0117 30 FC jr nc,fc1
0119 C6 A0 add a,GCOL
011B 26 FF ld h,0ffh
011D 24 fb1: inc h
011E D6 02 sub 2
0120 30 FB jr nc,fb1
0122 C6 02 add a,2
0124 6F ld l,a
; calcolo locazione video RAM
b1:
0125 AF xor a
0126 E5 push hl
0127 7A ld a,d
0128 21 EF80 ld hl,VEND
012B 01 0050 ld bc,COL
012E ED 42 b2: sbc hl,bc
0130 3D dec a
0131 20 FB jr nz,b2
0133 E3 ex (sp),hl
0134 4C ld c,h
0135 E3 ex (sp),hl
0136 09 add hl,bc
0137 E3 ex (sp),hl
0138 AF xor a
0139 BD cp l
013A E1 pop hl
013B 06 00 ld b,0
013D 28 12 jr z,b4

```

013F	B3		add	a,e	
0140	20 04		jr	nz,c1	
0142	CB C0		set	0,b	
0144	18 1B		jr	con	
0146	3D	c1:	dec	a	
0147	20 04		jr	nz,c2	
0149	CB D0		set	2,b	
014B	18 14		jr	con	
014D	CB E0	c2:	set	4,b	
014F	18 10		jr	con	
0151	B3	b4:	add	a,e	
0152	20 04		jr	nz,c3	
0154	CB C8		set	1,b	
0156	18 09		jr	con	
0158	3D	c3:	dec	a	
0159	20 04		jr	nz,c4	
015B	CB D8		set	3,b	
015D	18 02		jr	con	
015F	CB E8	c4:	set	5,b	
0161	7E	con:	ld	a,(hl)	
0162	87		add	a,a	
0163	30 03		jr	nc,d1	
0165	87		add	a,a	
0166	30 03		jr	nc,d2	
0168	3E 80	d1:	ld	a,80h	
016A	77		ld	(hl),a	
016B	7E	d2:	ld	a,(hl)	
016C	4F		ld	c,a	
016D	AF		xor	a	
016E	D1		pop	de	
016F	BB		cp	e	
0170	78		ld	a,b	
0171	28 03		jr	z,res	
0173	B1		or	c	
0174	18 02		jr	car	
0176	2F	res:	cpl		
0177	A1		and	c	
0178	77	car:	ld	(hl),a	
0179	C9		ret		
			end		

Macros:

Symbols:

A3	0113	B1	0125	B2	012E	B4	0151
C1	0146	C2	014D	C3	0158	C4	015F
CAR	0178	COL	0050	CON	0161	D1	0168
D2	016B	FB	010A	FB1	011D	FC	0102
FC1	0115	GCOL	00A0	GLIN	0048	RES	0176
START	0100	VEND	EF80				

No Fatal error(s)

Per dimostrarvi ancora piu' chiaramente l'uso della routine che vi abbiamo illustrato, vi forniamo un esempio tratto da quelli inviatici dal sig. Fabrizio Aversa. Il seguente programma disegna il grafico di una funzione che deve essere assegnata alla variabile F nella riga 80.

```
*****
* VERSIONE PER VIDEO 80 x 24 *
*****
```

```
10 FC=&AB00:GOSUB55000
20 ?CHR$(12)SPC(5)"GRAFICO FUNZIONI";
   ?"Plot della Funzione alla RIGA 80"
30 INTERRUPT1:INPUT"INTERVALLO ASSE X
   LIMITE INFERIORE";IP#:GOSUB50000:X1=IP:
   INPUT"LIMITE SUPERIORE";IP#:GOSUB50000:X2=IP
40 INPUT"INTERVALLO ASSE Y
   LIMITE INFERIORE";IP#:GOSUB50000:Y1=IP:
   INPUT"LIMITE SUPERIORE";IP#:GOSUB50000:Y2=IP
42 INTERRUPT0
44 ONERRORGOTO2000
50 DX=(X2-X1)/160
60 IY=(Y2-Y1)/72
70 DEFFNF(X)
75 ONERRORGOTO2000
80 F=SIN(X)/X
81 FNEND
82 PRINT CHR$(12)
83 IFX1>0THEN90
85 SX=-X1/DX+1
87 FORSY=1TO160:CALL FC,1,SX,SY:NEXTSY
90 IFY1>0THEN100
92 SY=-Y1/IY+1
94 FORSX=1TO160:CALL FC,1,SX,SY:NEXTSX
100 FORSX=1TO160
105 IFBYTEPOLL(0)=-1THENGOTO1000
106 D=FNF(X1+SX*DX)
110 SY=1/IY*(F-Y1)+1
115 IFSY<0ORSY>72THENNEXTSX
120 CALL FC,1,SX,SY
130 NEXTSX
1000 IFBYTE$(0)=" "THEN 20
1010 ?CHR$(12):END
2000 IFERL=80THENSX=SX+1:X=X+DX:RESUMENEXT
2010 POKE&E84F,&C5:GOTO1000
50000 IFLEFT$(IP#,1)="-"THEN50030
50010 IFLEFT$(IP#,2)<>"PI"THENIP=VAL(IP#):RETURN
50020 IP=PI/VAL(RIGHT$(IP#,(LEN(IP#)-3))):RETURN
50030 IFLEFT$(IP#,3)<>"-PI"THENIP=VAL(IP#):RETURN
50040 IP=-PI/VAL(RIGHT$(IP#,(LEN(IP#)-4))):RETURN
55000 OPEN #5,"I","set.com"
55010 FOR I=0 TO &79
55020 POKE FC+I,BYTE(#5)
55030 NEXT I
55040 CLOSE #5
55050 RETURN
```

