

# Sprite BASIC

Sprite BASIC ist eine Erweiterung des Locomotive BASIC um sechs weitere Befehle, die die Erzeugung und Steuerung von bis zu acht verschiedenen Sprites erlaubt. Die Befehle selbst sind keine RSX-Kommandos, sondern lassen sich, ähnlich unkompliziert wie die herkömmlichen BASIC-Befehle des CPC 464, einsetzen. Sprite BASIC befindet sich im oberen Ende des freien RAM-Speichers, belegt ca. 3 KB und ist voll Diskettenkompatibel, d.h., der unter AMSDOS reservierte Speicherbereich bleibt auch weiterhin unangetastet.

## Anweisung zum Erstellen einer schlüsselfertigen Version anhand der Listings:

Nachdem Sie das Programm LOADER abgetippt haben, speichern Sie es auf einen geeigneten Datenträger ab. Danach alles löschen, den CREATOR abtippen, starten und den erzeugten Maschinencode direkt hinter dem LOADER abspeichern. Zur Sicherheit den CREATOR auch auf einen Datenträger abspeichern. Darauf den Computer total zurücksetzen und den LOADER (am besten mit RUN) wieder einladen, jetzt erst mit dem Abtippen und Speichern der Programme DESIGNER und DEMO beginnen.

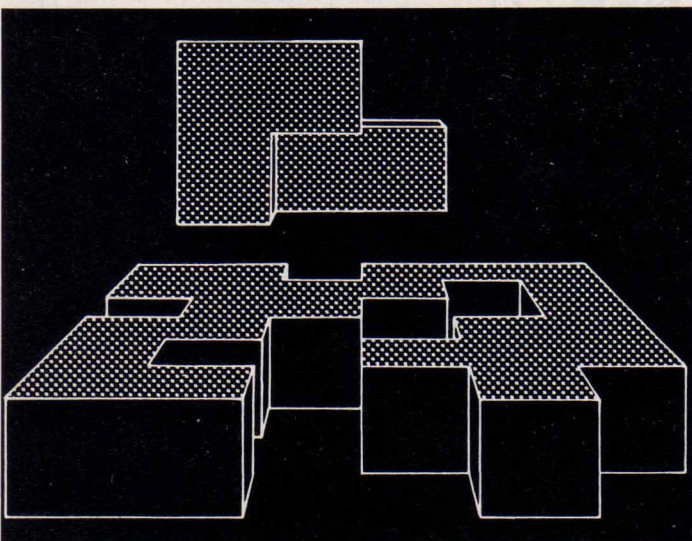
Um mit den neuen Befehlen arbeiten zu können, muß man immer erst den LOADER laden und ablaufen lassen. Er reserviert den nötigen Speicherplatz, verschiebt den Zeichensatz weiter nach unten und bindet das nachgeladene Maschinencodeprogramm in den BASIC-Interpreter des CPC 464 ein. Die Anwendung der Programme DEMO und DESIGNER ist nur mit vorher aktiviertem Sprite BASIC möglich. Ersteres erzeugt nacheinander 8 Sprites und bewegt sie zufallsgesteuert über den Bildschirm, wobei nach einer Kollision jeweils der Sprite verschwindet, der diese hervorgerufen hat. Der DESIGNER hingegen erlaubt die einfache Konstruktion der Sprites. Dies geschieht wahlweise über die Cursortasten oder einen Joystick.

Die Größe der einzelnen Sprites kann individuell festgelegt werden. Des weiteren spielt der Bildschirmmodus eine wichtige Rolle. Nachdem diese Eingaben für die gewünschten Sprites gemacht wurden, kommen wir zur eigentlichen Konstruktion der Sprites. Ein blinkender Cursor markiert die aktuelle Position, auf der gegenüber dem Original fünfmal vergrößerten Arbeitsfläche. Mit dem Feuerknopf oder der Copycursortaste werden die einzelnen Punkte gesetzt; mit DEL können sie wieder gelöscht werden. Die Funktionen der Tasten 'P' und 'I' entsprechen den PEN- und INK-Kommandos. Durch zweimaliges Drücken der ENTER-Taste beendet man diesen Arbeitsschritt und gelangt so zum nächsten Sprite. Danach kann man die so erzeugten Sprites abspeichern.

## Die Einzelbeschreibung der Befehle:

### DEFSPR s,b,h ...

Dieser Befehl erwartet mindestens drei Angaben: Die Sprite-



nummer (s:0-7), die Breite (b:1-8) und die Höhe (h:1-32) des zu erzeugenden Sprites. Die Breite wird in Bytes angegeben, da die Anzahl der Punkte in der Horizontalen abhängig vom jeweiligen Bildschirmmodus ist.

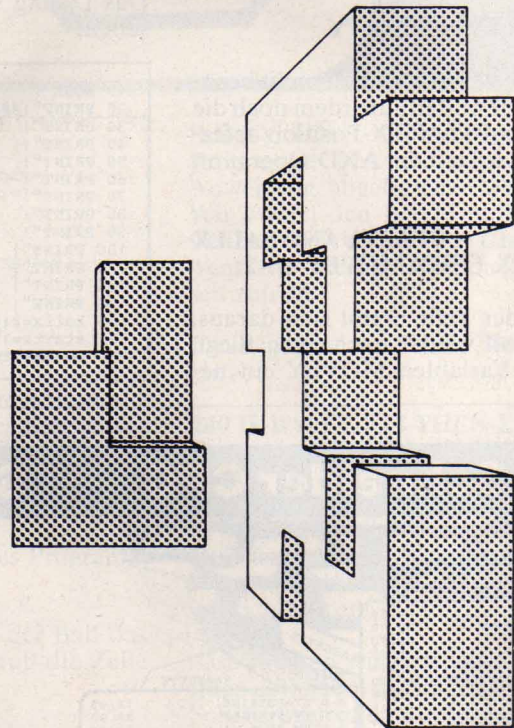
- Mode 0 Anzahl der Punkte pro Byte 2
- Mode 1 Anzahl der Punkte pro Byte 4
- Mode 2 Anzahl der Punkte pro Byte 8

Wenn man mehr als einen Sprite gleichzeitig definieren will, so kann man die dazu nötigen Parameter zusammen in einem DEFSPR-Befehl angeben.

Aktion: Abspeichern der angegebenen Dimensionen und Löschen des für die Spritematrix reservierten Speicherbereiches.

Die Werte von Breite und Höhe werden ab der Adresse 41820 in einem 32 Bytes großen Bereich abgelegt. Die jeweils 4 Bytes pro Sprite teilen sich wie folgt auf:

1. Byte: X-Koordinate
2. Byte: Y-Koordinate
3. Byte: Breite



```

10 '*****
20 '* *
30 '* Sprite BASIC *
40 '* (C)1985 *
50 '* by Thomas Fipl *
60 '* 6336 Solms/Lahn *
70 '* *
80 '*****
90 '
100 MEMORY 39999:CLS:FOR z=0 TO 51
110 PRINT CHR$(30)"Zeile:"200+z;
120 READ x$,p
130 FOR l=1 TO 32 STEP 2
140 m=VAL("&"MID$(x$,l,2))
150 POKE 39999+z*16+(l+1)/2,m
160 p=p XOR m:NEXT
170 IF p THEN PRINT"Error":STOP
180 NEXT:PRINT"ok"
190 SAVE"mcode",b,40000,832
200 DATA 11eeba180311f0bad5f79fcd67e7e37
3,101
201 DATA 2372e1c340dd3e08c3fbc1cd67ce4ff
7,253
202 DATA 2ccd67ce47c9cdd2a3e5cdf9a3782f4
7,24
    
```

203 DATA a677782ba677e1cd55dd38eac9f5cd0  
8,220  
204 DATA a47ea02805f1f5cd1fa4f1c9dd215ca  
3,30  
205 DATA 21ecbaf58787dd85dd6fc104af37171  
0,50  
206 DATA fd47c95fdd7e023dfe08d0dd7e033df  
e,245  
207 DATA 20d07bdd4dc5e5f3d9215c9b846779e  
e,131  
208 DATA 0ced79c5dd4602dd4e03d9fd21e6a42  
1,50  
209 DATA c8b17e36005f3257a4fd7e00fd6bd5  
6,49  
210 DATA 01dd5e00dd6fd53ec7926faffd67576  
7,228  
211 DATA cda90bd17afec83054e5d5d91600c5c  
5,105  
212 DATA 7e234fd94b1c1ccb393024d9dda5fd5  
d,103  
213 DATA cb3b380428030f0f0f0fb247dd7d2fa  
1,60  
214 DATA fd5dcb3b38042803070707075778d9c  
d,124  
215 DATA d8a4cdf90bd9c110c6c17ad94bcb39d  
c,138  
216 DATA d8a4d1e1cd130c14d90dd920a7fd7d3  
2,178  
217 DATA c8b1c1d9e1c1dd69fffffc94779fe5  
0,83  
218 DATA d078ae77b8c8fd26ffc9aacff0c5cdf  
9,139  
219 DATA a3d1dd7201dd7300cd1fa47eb077237  
8,230  
220 DATA 2fa64f78fda4b1772bc9cdd2a3f5cd5  
5,18  
221 DATA dd3021cdd7a3e37ccde9a44f782f472  
b,30  
222 DATA a6772b7ea07779e1a7c8ed5bf0ba7ab  
3,145  
223 DATA c2f3c6c9c148cd17a43aebbaa0c879e  
5,86  
224 DATA f58787f587217ca3856f22a4ac4e234  
6,146  
225 DATA c5234e2346235e2356237e23666ff1f  
5,182  
226 DATA 8338045fba3804925f444de1f155215  
c,154  
227 DATA a3856f7e23666f78844779854ff1d5c  
d,252  
228 DATA e9a4d14f782b2b15280c2fa6772aa4a  
c,70  
229 DATA 722373791891b67723782fa677e1ed5  
b,225  
230 DATA eeba188acdd2a3f5f72ccdd7a3f1e5f  
5,12  
231 DATA 8787f5215ca3856f7e23666f1100007  
8,252  
232 DATA 9428073004ed4415151447799528073  
0,134  
233 DATA 04ed441d1d1c4fe1b8300669d51e001  
8,43  
234 DATA 066841d516004d7c87217ca3856f702  
3,29  
235 DATA 781f77237323722371237023d173237  
2,192  
236 DATA c1cd17a421eaba2fa677237eb077e1c  
d,134  
237 DATA 55dd3890c9cdd2a3f5f72ccd27c37af  
e,214  
238 DATA 0830037bfe20d205c2e37c4f0600878  
7,93  
239 DATA 215ca3856f70237023141c722373792  
1,70  
240 DATA 5c9b8467cd4fa6e1cd55dd38c8c921e  
a,86  
241 DATA ba0608af772310fcc9d6bafef0bd0c1e  
b,9

## MICA

### Das CAD-Programm der Zukunft, in Preis und Leistung.

- Superleichte Bedienung
  - Symbolbibliotheken für Elektrotechnik und Layouterstellung sind bereits integriert, weitere können erstellt werden
  - Symbole können verkleinert, vergrößert, gedreht und gespiegelt werden
  - Eine Arbeits- und 5 zusätzliche Ebenen stehen dem Benutzer stets zur Verfügung
  - Alle Ebenen sind gleichzeitig auf dem Bildschirm, Drucker oder Plotter darstellbar
  - Zeichnungen jeder Art, Layouts usw. sind einfachst zu erstellen
  - Maßstabgerechtes Konstruieren in Millimeter oder Zoll
  - Zoomfunktion für die Bildarstellung
  - Rasterfunktion
  - Ausdruck auf Drucker und Plotter maßstabgetreu, verkleinert oder vergrößert, beim Plotten auch in Farbe
  - MICA ist in deutsch, mit ausführlichem Handbuch
  - MICA ist lieferbar:
    - alle Schneider Computer mit Floppy
    - IBM-PC und Compatible
    - PMS 88
    - MC-Computer mit Term I
  - MICA ist in Vorbereitung für:
    - Atari 520 ST
    - Apple II und Compatible
    - Commodore 128
  - MICA wird geliefert für 198,- DM per Nachnahme zuzügl. Versandkosten (5,- DM) oder nach Vorkasse durch V-Scheck frei Haus, in das Ausland nur Vorkasse
- CP/M ist das eingetragene Warenzeichen von Digital Research.  
PC-DOS ist das eingetragene Warenzeichen von IBM.  
Bitte geben Sie System, Drucker an.  
Dies ist deshalb notwendig, da es so viele Printer gibt, und eine Druckerauswahl aus Speicherplatzgründen nicht mehr implementierbar ist.

läuft unter CP/M 2.2  
läuft unter PC-DOS  
läuft unter CP/M 86  
läuft unter CP/M 2.2

**Alleinvertreib:**  
**E & C Rupert Zellmeier**  
**Dompfaffstraße 127a**  
**8520 Erlangen**

## CIZ COMPUTER-INFO-ZENTRUM

\* einziges, autorisiertes Schulungszentrum  
in Bayern für SCHNEIDER-Computer

\* Seminare: ComPack, TexPack  
Jeden Donnerstag, Freitag

\* Schnellkurse Computer: jeden Samstag  
zum Sonderpreis

Fordern Sie Beschreibung, Termine+Preise an!

\* NEU: 1 MB-Floppy für Schneider Computer

CPC 464	DM 659,--
CPC 664	DM 799,--
CPC 6128	DM 799,--

\* DRUCKER 100 Zeichen/Sek. TAXAN CPA 80 GS  
für Schneider Computer  
echte Near Letter Quality  
DM 799,--

\* Commodore 128 Preis a.Anfrage sof.lieferb.  
Mehr Info über Hard- und Software bei  
Ihrem Spezialisten:

**CIZ - Computer Info Zentrum**  
**Dr. Naumann Elektronik GmbH**

Hofrat-Röhler-Str.7  
8900 Augsburg  
Tel. 0821-555011  
täglich 9-17 Uhr

Veit-Stoss-Str.50  
8000 München 21  
Tel. 089-587533/5808041

# Programme

## 4. Byte: Höhe

Die reservierten Bereiche für die Spritematrizen zählen jeweils 256 Bytes und beginnen bei der Adresse 39772.

### SPRITE s,x,y

Zeichnet den Sprite mit der Nr. 's' an die angegebene Position (x,y), beginnend bei der linken oberen Ecke des Sprites. Die Koordinaten sind modeunabhängig, haben ihren Nullpunkt in der linken oberen Ecke des Bildschirms und betragen in waagerechter Richtung 0-169 und senkrecht 0-199.

### CLSPR s ...

Dieser Befehl löscht einen, bei entsprechender Anzahl der Parameter auch mehrere Sprites.

### MOVSPR s,x,y ...

Definiert für Sprite 's' einen Zielpunkt (x,y), der dann, z.B. in-terruptgesteuert, mit dem Befehl SPRITE s (ohne Koordinatenangaben!) auf dem direkten Weg angesteuert werden kann. Auch hier können gleichzeitig mehrere Sprites mit Zielkoordinaten versehen werden.

### ON CRASH GOSUB z

Einmal im Programm gesetzt, verursacht jede Überlagerung eines Zeichens oder einer Grafik mit einem Sprite einen Sprung in ein Unterprogramm, beginnend bei der Zeile z. Hier kann dann anhand des Wertes in der Speicherzelle 47853 festgestellt werden, welcher Sprite der Verursacher war. Dieser Bytewert ist bitsignifikant, d.h., jedes Bit, angefangen von rechts, steht stellvertretend für einen Sprite. Um einen einzelnen Sprite zu testen, schreibt man am besten:

... IF PEEK(a) AND s IS THEN ...

a ist die Adresse und s ist die Nummer des zu testenden Sprites.

### ON AIM GOSUB z

Mit Hilfe dieses Befehls kann man feststellen, ob ein Sprite bereits seinen Zielpunkt (siehe MOVSPR) erreicht hat, und ihm in dem zugehörigen Unterprogramm, z.B. einen neuen Zielpunkt zuweisen. In der Speicherzelle 47850 steht der bitsignifikante Wert, mit dem festgestellt werden kann, welcher Sprite seinen Zielpunkt erreicht hat.

### Auflistung der neu hinzugekommenen Systemvariablen:

(Die Befehle in Klammern beeinflussen die Flagregister)

#### 47850

Zielpunkt erreicht?

(SPRITE s, SPRITE s,x,y=0, MOVSPR=0)

#### 47851

Zielpunkt definiert?

(MOVSPR=1, SPRITE s,x,y=0, CLSPR=0)

#### 47852

Sprite abgebildet?

(SPRITE=1, CLSPR=0)

#### 47853

Überlagerung?

(SPRITE)

#### 47854/55

Zeilenadresse von ON AIM GOSUB

#### 47856/57

Zeilenadresse von ON CRASH GOSUB

Um die automatischen Sprungbefehle innerhalb eines Programms wieder rückgängig machen zu können, genügt es, die entsprechenden Speicherzellen mit zwei POKE-Befehlen auf 0 zu setzen. Mit Erscheinen der 'Ready'-Meldung wurden alle Systemvariablen von Sprite BASIC wieder gelöscht, so daß der Einsatz der neuen Befehle im Direktmodus unter Umständen nicht das gewünschte Resultat erbringt.

Thomas Fippel

```
242 DATA c66426a66fc3bbdd06a5e2a3a0a511a
6,184
243 DATA c1a3bca3e511d6a6eb0edd1acd8affb
e,221
244 DATA 280c7e231730fb0c7ea720efe1c9231
3,167
245 DATA 7ee67f471acd8affb82015be28f0eb2
3,33
246 DATA 7ecd7bff3809f1f1f179d1c1c383dfe
b,234
247 DATA d1d518cefedd3804fee33802b7c9333
3,148
248 DATA 23f5e521d6a6d6dd280af57e231730f
b,63
249 DATA f13d20f6cd1ae2c337e25350524954c
5,176
250 DATA 434c5350d24d4f565350d2444546535
0,31
251 DATA d24f4e2043524153c84f4e204149cd0
0,220
```

### Loader

```
100 SYMBOL AFTER 256:MEMORY 39771
110 POKE &AE7D,91:POKE &AE7E,155
120 POKE &B296,91:POKE &B297,155
130 CLEAR:SYMBOL AFTER 240
140 LOAD"!mcode",41916
150 FOR n=41820 TO 41851
160 POKE n,0:NEXT
170 FOR n=1 TO 5:READ a,b
180 POKE a,&C3:POKE a+1,b AND 255
190 POKE a+2,INT(b/256)AND 255
200 NEXT:BORDER 2:MODE 1:LOCATE 3,2
210 PRINT"Sprite BASIC 1.0 ";
220 PRINT"39273 Bytes free",,:NEW
230 DATA &ac01,&a64a,&ac07,&a655
240 DATA &ac13,&a6b0,&ac16,&a670
250 DATA &30,&dd37
```

### Designer

```
100 CLEAR:DEFSTR a-k:DEFINT m-z
110 FOR n=300 TO 313:READ m:POKE n,m:NEX
T
120 DATA &dd,&66,3,&dd,&6e,2,&cd
130 DATA &26,&bc,&eb,&73,&23,&72,&c9
140 FOR n=1 TO 8:READ m:k=k+CHR$(m):NEXT
150 DATA 240,241,242,243,224,127,112,105
160 KEY DEF 72,1,240,240,240
170 KEY DEF 73,1,241,241,241
180 KEY DEF 74,1,242,242,242
190 KEY DEF 75,1,243,243,243
200 KEY DEF 76,1,224,224,224
210 DIM prm(7,3):BORDER 2:SPEED KEY 12,1
220 SYMBOL 255,248,248,248,248,248,0,0,0
230 MODE 1:WINDOW#1,12,27,8,12:LOCATE 11
,3
240 PRINT CHR$(24)" Sprite Designer "CHR
$(24)
250 FOR n=0 TO 7:LOCATE 11,6
260 PRINT"Sprite"n"(j/n)? "CHR$(143);
270 i=LOWER$(INKEY$):IF i=""GOTO 270
280 IF i="j"THEN PRINT CHR$(8)i;:GOSUB 3
00
290 NEXT:IF sp GOTO 370 ELSE END
300 LOCATE 12,8:INPUT"Mode (0-2) ";prm(
sp,1)
310 LOCATE 12,10:INPUT"Breite(1-8) ";prm
(sp,2)
320 LOCATE 12,12:INPUT"Hoehe (1-32)";prm
(sp,3)
330 prm(sp,0)=n:sp=sp+1:CLS#1:RETURN
340 DI:MOVE 16+(px-1)*t*5,361-py*10
350 TAG:PRINT CHR$(255);:TAGOFF
360 v=v XOR 1:EI:RETURN
370 FOR z=0 TO sp-1:DIM mtx(31,63)
380 MODE prm(z,1):PRINT CHR$(23)CHR$(1)
390 t=2^(2-prm(z,1)):s=2^(1+prm(z,1))
400 x=t*s*prm(z,2)*5+2:y=10*prm(z,3)+2
```

## Programme

```

410 MOVE 15,352:DRAWR x,0,1:DRAWR 0,-y
420 DRAWR-x,0:DRAWR 0,y:r=s*6
430 LOCATE r,9:PRINT"Spr.:"prm(z,0)
440 LOCATE r,10:PRINT"Mode:"prm(z,1)
450 LOCATE r,11:PRINT"Br.:"prm(z,2)
460 LOCATE r,12:PRINT"H.:"prm(z,3)
470 LOCATE r+1,14:PRINT CHR$(240)" DEL
"
480 LOCATE r,15:PRINT CHR$(242)"c"CHR$(2
493)
490 LOCATE r+1,16:PRINT CHR$(241)" ENTE
R"
500 LOCATE R,18:PRINT"P - Pen"
510 LOCATE r,19:PRINT"I - Ink"
520 px=1:py=1:pin=1
530 PLOT-9,-9, pin:EVERY 15 GOSUB 340
540 LOCATE s,2:PRINT"xpos"px
550 LOCATE s+9,2:PRINT"ypos"py
560 i=LOWER$(INKEY$):IF i=""GOTO 560
570 m=REMAIN(0):IF v THEN GOSUB 340
580 ON INSTR(k,i)GOSUB 670,690,710,730,7
590,780,800,810
590 IF i<>CHR$(13)GOTO 530
600 LOCATE r-1,24:PRINT"beenden ?"
610 i=INKEY$:IF i=""GOTO 610
620 LOCATE r-1,24:PRINT SPC(9)
630 IF i<>CHR$(13)GOTO 530
640 GOSUB 830:NEXT:MODE 1
650 LOCATE 2,2:INPUT"Dateiname";d
660 SAVE d,b,39772,2080:END
670 IF py>1 THEN py=py-1
680 GOTO 340
690 IF py<prm(z,3)THEN py=py+1
700 GOTO 340
710 IF px>1 THEN px=px-1
720 GOTO 340
730 IF px<s*prm(z,2)THEN px=px+1
740 GOTO 340
750 PLOT 383+px*t,353-py*2,mtx(py-1,px-1
)
760 GOSUB 340:PLOT 383+px*t,353-py*2, pin
770 mtx(py-1,px-1)=pin:GOTO 340
780 PLOT 383+px*t,353-py*2,mtx(py-1,px-1
)
790 v=1:mtx(py-1,px-1)=0:GOTO 340
800 LOCATE r,21:INPUT"Pen";pin:RETURN
810 LOCATE r,22:INPUT"ink";m,n
820 INK m,n:RETURN
830 p=&C11F:l=41820+4*prm(z,0)
840 POKE l,0:POKE l+1,0
850 POKE l+2,prm(z,2)
860 POKE l+3,prm(z,3)
870 l=39772+256*prm(z,0)
880 FOR y=1 TO prm(z,3)
890 FOR x=1 TO prm(z,2)
900 POKE l,PEEK(p+x):l=l+1
910 NEXT:CALL 300,p,@p:NEXT
920 ERASE mtx:RETURN

```

### Demo

```

100 MODE 0:BORDER 2:DEFINT m,n,x,y
110 DEF FNs(a)=LOG(PEEK(a))/LOG(2)
120 DEF FNx=INT(RND*160)
130 DEF FNy=INT(RND*200)
140 FOR n=0 TO 7:DEFSPR n,4,16
150 READ b:m=UNT(39772+256*n)
160 FOR m=m TO m+63:POKE m,b:NEXT
170 SPRITE n, FNx, FNy:MOVSPR n, FNx, FNy
180 NEXT:DATA 89,76,34,98,45,67,52,41
190 ON AIM GOSUB 260
200 ON CRASH GOSUB 240
210 EVERY 10 GOSUB 230
220 GOTO 220
230 FOR n=0 TO 7:SPRITE n:NEXT:RETURN
240 SOUND 1,100,1,2:CLSPR FNs(47853)
250 POKE 47853,0:RETURN
260 MOVSPR FNs(47850), FNx, FNy:RETURN

```

## PROBLEME, FRAGEN ANREGUNGEN...?

JEDEN MITTWOCH ZWISCHEN 14.00 UND 17.00 UHR  
STEHEN IHNEN UNSER HERR RITTER (REDAKTION) UND  
HERR MORGEN (PROGRAMMIERUNG) AM



## HEISSEN DRAHT

ZUR VERFÜGUNG. RUFEN SIE DOCH EINFACH AN.

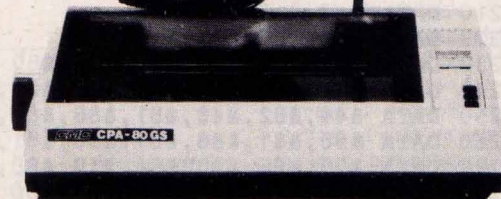
**Tel.: 05651/8702**

## Drucker CPA- 80 GS,

100 Zeichen/sek.,  
DIP-Schalter au-  
ßen, SCHNEIDER- auf EPSON-  
Kompatibilität umschaltbar, down-  
load, 9 nationale Zeichensätze,  
Plott-Funktion, d. h. software-  
mäßig Papiervorschub vor- oder  
rückwärts, 9 Nadeln. Das 8. BIT wird  
auf diesem Drucker simuliert, da-  
her voll grafikfähig!

incl. Anschlußkabel

**DM 798,-**



COMPUTERHANDELS GmbH

Hildesheimer Straße 52  
D-3000 Hannover 1  
Tel: (0511) 88 78 40  
(0511) 88 27 09  
Telex: 9218155 ies d