

40: legt
Wenn Sie nachzählen wollen, es stehen DATA für 64 neu gestaltete Zeichen zur Verfügung. Das Programm TYPWEX64.B12 lädt aber, wie besprochen, nur 63 davon, denn das nullte Zeichen bleibt als Leerzeichen erhalten. Wenn Sie die FOR-NEXT-Schleife auf 0 TO 63 ändern, dann können Sie sehen, was es bedeutet, kein Leerzeichen zu haben. Der Trickfilm ist dann zwar um ein Bild länger, aber der ganze übrige Bildschirm ist voll von gestalteten "Leerzeichen".

3.2.3 Character Dump

Wenn Aufbau und Lage des Zeichensatz-Speichers bekannt sind, dann ist es kein Problem mehr, die Daten eines bestimmten Zeichens zu lesen und in irgendeiner Weise weiter zu verarbeiten. So ist es z.B. möglich, die ATARI-Standard-Zeichen in ihren original Bit-Mustern mit dem Nadeldrucker auf Papier zu bringen. Schwierigkeiten bereitet höchstens die Kommunikation zwischen beiden Geräten. Im Kapitel 4 wird darüber ausführlicher zu reden sein.

Das markanteste Problem bei der Koordination von Rechner und Drucker ist die Tatsache, daß der Computer die Bit-Muster der Zeichen Zeilenweise zu Dezimalwerten aufrechnet, der Drucker aber spaltenweise arbeitet.

Es gibt verschiedene Wege, dieses Problem zu lösen. Am simpelsten ist es, dem Drucker die Daten aus dem Computer so zu senden, wie sie dort liegen. Die Zeichen werden dann um 90° gedreht auf das Papier gedruckt und wenn sie nicht spiegelverkehrt erscheinen sollen, müssen die Daten rückwärts verarbeitet werden. Wie ein solches Programm im Prinzip aufgebaut ist, hat Ihnen DEFCHRPR.B12 bereits gezeigt. Sollen die Zeichen aber in richtiger Orientierung auf dem Blatt enden, dann ist ein etwas aufwendigeres Programm notwendig, das die zeilenweisen Daten-Byte aus dem Zeichensatz-Speicher des Rechners in Bits zerlegt und spaltenweise neu zu Daten-Bytes zusammenrechnet, die dem Drucker passen;


```

0 REM CHRPR.T.B12
100 LPRINT CHR$(27);CHR$(68);CHR$(4);CHR$(7);CHR$(10)
;CHR$(13);CHR$(16);CHR$(19);CHR$(22);CHR$(25);CHR$(0)
;
110 Q=6
200 DIM X(7,7)
210 FOR INV=0 TO 1
220 FOR ATA=0 TO 127
230 INC=ATA
240 IF ATA<32 THEN INC=ATA+64:GOTO 260
250 IF ATA<96 THEN INC=ATA-32:GOTO 260
260 FOR M=0 TO 7:FOR N=0 TO 7:X(M,N)=0:NEXT N:NEXT M
290 FOR I=0 TO 7
300 A=PEEK(57344+I+INC*8)
310 IF A>127 THEN A=A-128:X(0,I)=1
320 IF A>63 THEN A=A-64:X(1,I)=1
330 IF A>31 THEN A=A-32:X(2,I)=1
340 IF A>15 THEN A=A-16:X(3,I)=1
350 IF A>7 THEN A=A-8:X(4,I)=1
360 IF A>3 THEN A=A-4:X(5,I)=1
370 IF A>1 THEN A=A-2:X(6,I)=1
380 IF A>0 THEN X(7,I)=1
390 NEXT I
400 A=X(0,0)*128+X(0,1)*64+X(0,2)*32+X(0,3)*16+X(0,4)
*8+X(0,5)*4+X(0,6)*2+X(0,7)
410 A=ABS(INV*255-A)
450 B=X(1,0)*128+X(1,1)*64+X(1,2)*32+X(1,3)*16+X(1,4)
*8+X(1,5)*4+X(1,6)*2+X(1,7)
460 B=ABS(INV*255-B)
500 C=X(2,0)*128+X(2,1)*64+X(2,2)*32+X(2,3)*16+X(2,4)
*8+X(2,5)*4+X(2,6)*2+X(2,7)
510 C=ABS(INV*255-C)
550 D=X(3,0)*128+X(3,1)*64+X(3,2)*32+X(3,3)*16+X(3,4)
*8+X(3,5)*4+X(3,6)*2+X(3,7)
560 D=ABS(INV*255-D)
600 E=X(4,0)*128+X(4,1)*64+X(4,2)*32+X(4,3)*16+X(4,4)
*8+X(4,5)*4+X(4,6)*2+X(4,7)
610 E=ABS(INV*255-E)
650 F=X(5,0)*128+X(5,1)*64+X(5,2)*32+X(5,3)*16+X(5,4)
*8+X(5,5)*4+X(5,6)*2+X(5,7)
660 F=ABS(INV*255-F)
700 G=X(6,0)*128+X(6,1)*64+X(6,2)*32+X(6,3)*16+X(6,4)
*8+X(6,5)*4+X(6,6)*2+X(6,7)
710 G=ABS(INV*255-G)
750 H=X(7,0)*128+X(7,1)*64+X(7,2)*32+X(7,3)*16+X(7,4)
*8+X(7,5)*4+X(7,6)*2+X(7,7)
760 H=ABS(INV*255-H)

```



```

800 OPEN #1,8,0,"P:"
810 Q=Q+1:IF Q=7 THEN PUT #1,10:Q=-1:GOTO 830
820 FOR T=0 TO Q:PUT #1,9:NEXT T
830 PUT #1,27
840 PUT #1,75
850 PUT #1,8
860 PUT #1,0
870 PUT #1,A
880 PUT #1,B
890 PUT #1,C
900 PUT #1,D
910 PUT #1,E
920 PUT #1,F
930 PUT #1,G
940 PUT #1,H
950 CLOSE #1
960 NEXT ATA
970 NEXT INV

```

100: setzt die Tabulator-Marken des Druckers. Das Format dieser Anweisung ist auf den radix-10 von star abgestellt. Falls Sie einen anderen Drucker haben sollten, müssen Sie diese Zeile entsprechend ändern. (Es werden übrigens nur sechs TAB-Marken benötigt.)

110: Die Variable Q bekommt einen Anfangswert, der dafür sorgt, daß vor dem Ausdruck des ersten Zeichens in Zeile 810 bereits ein Zeilenvorschub an den Drucker gesendet wird. Der automatische Zeilenvorschub bei Empfang des Signals CR (carriage return = RETURN-Taste) muß unterdrückt sein (star: DIP-switch C-4: OFF).

210: Das Programm muß zweimal abgearbeitet werden, um einen normalen und einen inversen Zeichensatz auszudrucken.

220: Die folgende Schleife arbeitet 128 Werte ab.

240 und 250: verwandeln die ATASCII-Werte in internen Code.

290 bis 390: Die acht Daten-Byte des bearbeiteten Zeichens werden gelesen und in ihr Bit-Muster aufgespalten. Die doppelt-indizierte Variable X(n,n) nimmt die acht mal acht Bit-Werte auf.

400 bis 760: Die in X(n,n) erfaßten Bit-Muster werden hier spaltenweise zu Dezimalwerten aufgerechnet. Die Variable INV bewirkt, daß die Zeichen normal oder invers gedruckt werden. Ist INV=0, dann bleibt der Multiplikator 255 in der Klammer ohne Wirkung und das Minuszeichen wird durch die ABS-Funktion unschädlich gemacht. Ist aber INV=1, dann

800: Der Schreib-Kanal zum Drucker wird eröffnet.

820: besorgt die Tabulator-Sprünge.

```
870 bis 940: senden acht Bit-Muster an den Drucker.
```

♥	†		∫	÷	π	/	\
▲	•	◀	■	■	—	—	■
♣	°	—	+	●	≡		T
⊥		L	E	+	+	+	+
()	"	#	\$	%	&	'
@	1	2	3	,	-	.	/
B	9	:	;	4	=	>	?
H	A	B	C	D	E	F	G
P	I	J	K	L	M	N	O
X	Q	R	S	T	U	V	W
•	a	b	c	\	j	^	_
h	i	j	k	d	e	f	g
p	q	r	s	l	n	v	o
x	y	z	•	t	w	4	p

A 10x10 grid of 100 small, stylized, black and white images, each representing a different character or symbol from a non-Latin alphabet, likely a Cyrillic or Greek alphabet, arranged in a grid pattern.