

# EGA.CPI

## Historische Bemerkungen zu PC-Bitmap-Schriften



Man unterscheidet zwischen skalierbaren Vektorschriften (z.B. TrueType, PostScript Type 1) und Bitmap-Schriften (bzw. Matrix-Schriften). Bitmap-Schriften gab es (und gibt es) sowohl für Computerbildschirme (Monitore) als auch für Drucker (Nadel- bzw. Matrixdrucker).

Letztlich setzt sich jede Schrift am Bildschirm eines Computers aus Bildpunkten zusammen, wie man leicht mit der Lupe prüfen kann, doch zu DOS-Zeiten gab es nur Bitmap-Schriften, bei denen jedes Zeichen in eine starre Matrix mit z.B. 8 x 8, 8 x 14 oder 8 x 16 Bildpunkten (Breite x Höhe, d.h. die Breite umfaßte immer 8 Bildpunkte, die Höhe war dagegen variabel) gepreßt wurde, wobei jedem gesetzten Bit (Einser-Bit) ein sichtbarer Bildpunkt entsprach.

Die Bitmap-Schrift für den 437-CP-Zeichensatz befand sich (und befindet sich heute noch) im ROM des PC. Dieser ROM-Zeichensatz wird beim Booten benutzt (solange sich Windows noch nicht im Speicher befindet) und wenn man das Hardware-Konfigurations-Menü aufruft. Der 437-CP-Zeichensatz in interner 8 x 16 Matrix ist unten auf den Seiten 7–14 abgebildet.

Darüber hinaus konnte man früher (und könnte auch heute noch) mittels CONFIG.SYS und AUTOEXEC.BAT die Bitmap-Schriftendatei EGA.CPI (EGA = Enhanced Graphics Adapter, CPI = Code Page Information) in den RAM-Speicher des PC laden, um von der zuerst durch das ROM vorgegebenen 437-Code Page zu einer anderen Code Page wechseln zu können. Eine Darstellung verschiedener anderer Code-Pages für Windows und Macintosh findet sich in unserem Handbuch <http://www.sanskritweb.net/fonts/pahotman.pdf>, Seiten 34-37.

Die Datei EGA.CPI gibt es auch heute noch unter Windows XP, und die darin enthaltenen Bitmap-Schriften werden benutzt, wenn über die Windows-DOS-Box alte DOS-Programme aufgerufen werden. Siehe z.B. obigen Windows-XP-Screendump des DOS-Edit-Programms, das hier zu Demonstrationszwecken auf die auch bei Windows XP noch verfügbare extrem grobe Matrix mit 8 x 8 Bildpunkten eingestellt wurde.


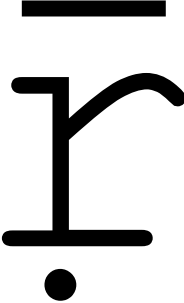
Wenn früher z.B. das aus 8 x 8 Bildpunkten bestehende A am Bildschirm angezeigt wurde, so geschah dies derart, daß die 8 Bytes mit dem Bitmuster des A in den Bildschirmspeicher durch maschinensprachlichen Befehl kopiert und auf diese Weise als leuchtende Bildpunkte auf dem Bildschirm zur Anzeige gebracht wurden.

# Historisches Anwendungsbeispiel

Früher, als es noch kein Windows gab, war die Darstellung seltener diakritischer Zeichen nur möglich, falls man damals eine der in der Datei EGA.CPI enthaltenen 437-Codepage-Zeichensätze modifizierte. Als ich z.B. am Ende des Jahres 1988 damit anfang, die 1. Auflage meines Sanskrit-Kompendiums zu schreiben (siehe <http://www.sanskritweb.net/deutsch>), war an Devanagari-Schrift (siehe <http://www.sanskritweb.net/itrans>) noch nicht zu denken. Selbst die Darstellung der 15 Sonderzeichen zur Transkription des Sanskrit, nämlich

ā ī ū ṛ ṝ ḷ ṅ ñ ṇ ṭ ḍ ś ṣ ṃ ḥ

war am Bildschirm nur nach dem "Patchen" (d.h. Modifizieren) der Datei EGA.CPI möglich. Zu diesem Zweck schrieb ich ein Programm namens **EGA-KON1.BAS** (siehe unten Seite 3), das den in der Datei EGA.CPI enthaltenen 437-Zeichensatz in eine Textdatei umwandelte, die ich dann gemäß der obigen Transliterationszeichen modifizierte (siehe unten Seite 13, blaue Matrix-Zeichen), und danach mit dem Programm **EGA-KON2.BAS** in eine modifizierte ("gepatchte") EGA.CPI-Datei zurückspeicherte.

<pre> 230 - ṝ ..... ..... ..... #####. ..... #.###. .###.##. .##.##. .##..... .##..... .##..... .##..... #####. ..... .##..... .##..... .....                 </pre>	<pre> 230 - ṝ                 </pre> 	<pre> 230 - ṝ                 </pre> 
8 x 16 Matrix in Textdatei	8 x 16 Bildschirm-Matrix	Ideallösung (Vektorfont)

Man muß die obige Bildschirm-Matrix (und auch die Muster unten auf den Seiten 7 bis 13) mit großem Leseabstand betrachten, um die Schriftzeichen überhaupt erkennen zu können.

Ulrich Stiehl, Dezember 2006

## EGA-KON1.BAS

```
REM Umwandlung von ega.cpi in Sanskrit-Transliteration/5.3.1989/us
REM =====

REM EGA-KON1.BAS
REM =====
REM
COLOR 7, 1: CLS
PRINT "EGA-KON1.BAS"
PRINT "=====
PRINT
PRINT "1. Schritt: Umwandlung von EGA-ALT.CPI in EGA-ALT.ASC"
PRINT "          Herausfiltern der 256 8x16-Zeichen"

REM Struktur von Original-EGA.CPI = EGA-ALT.CPI
REM =====

REM 65 Bytes Header - siehe DOS Technical Reference, S. 7-17
REM 256 x 16 Bytes für 8x16-Ega-Font = 4096 Bytes
REM Danach weitere Fonts - hier nicht geändert

REM Binärtabelle für Byte
REM =====

DIM bin%(7)
FOR x% = 0 TO 7: bin%(x%) = 2 ^ x%: NEXT x%

OPEN "ega-alt.cpi" FOR BINARY AS #1
OPEN "ega-alt.asc" FOR OUTPUT AS #2

REM 65-Byte Header
REM =====

puffer$ = "1": REM Länge 1
FOR x = 1 TO 65: GET #1, , puffer$: NEXT x

REM 8x16-Byte-Zeichen
REM =====

puffer$ = "0123456789ABCDEF": REM Länge 16

FOR zeichen% = 0 TO 255
  GET #1, , puffer$
  header$ = STR$(zeichen%)
  IF zeichen% > 31 AND (zeichen% <> 127 AND zeichen <> 255) THEN
    header$ = header$ + " - " + CHR$(zeichen%)
  END IF
  PRINT #2, MID$(header$, 2)
  GOSUB hextobin
NEXT zeichen%
CLOSE

PRINT
PRINT "a) ASCII-Datei 'EGA-ALT.ASC' in F&A-Datei 'EGA-NEU.FA' importieren"
PRINT
PRINT "b) Bitmap ändern und EGA-NEU.FA zwischenspeichern"
PRINT
PRINT "c) F&A-Datei 'EGA-NEU.FA' in ASCII-Datei 'EGA-NEU.ASC' exportieren"

END

REM 16 Bytes in 16 8-Bit-Zeilen
REM =====

hextobin:
```

```

FOR h1% = 1 TO 16
  hexstring$ = ""
  h$ = MID$(puffer$, h1%, 1)
  h$ = h$ + CHR$(0)
  h% = CVI(h$)
  FOR h2% = 7 TO 0 STEP -1
    IF (h% AND bin%(h2%)) = 0 THEN
      hexstring$ = hexstring$ + "."
    ELSE
      hexstring$ = hexstring$ + "#"
    END IF
  NEXT h2%
  PRINT #2, hexstring$
NEXT h1%

RETURN

```

### **EGA-KON2.BAS (inkl. EGA-KON3.BAS)**

```

REM Umwandlung von ega.cpi in Sanskrit-Transliteration/5.3.1989/us
REM =====
COLOR 7, 1: CLS
PRINT "EGA-KON2.BAS"
PRINT "======"
PRINT
PRINT "2. Schritt: Umwandlung von EGA-NEU.ASC in EGA-NEU.BIN"

REM Struktur von EGA-NEU.ASC
REM =====

REM 256 x 17 Zeilen: 1.      Zeile Kommentar
REM                      2.-17. Zeile ASCII-Bit-Muster

REM Beispiel Bitzeile: "..##..##"
REM # = Bildpunkt gesetzt
REM . = Bildpunkt nicht gesetzt

REM Binärtabelle für Byte rückwärts
REM =====

DIM bin%(7)
y% = 7
FOR x% = 0 TO 7
  bin%(y%) = 2 ^ x%
  y% = y% - 1
NEXT x%

REM Testweise lesen
REM =====

OPEN "ega-neu.asc" FOR INPUT AS #1
counter% = 0
WHILE NOT EOF(1)
  LINE INPUT #1, x$
  x% = VAL(x$)
  IF x% = 0 AND LEFT$(x$, 1) <> "0" THEN
    PRINT "Falsche Datei"
    CLOSE : END
  END IF
  FOR x% = 1 TO 16
    LINE INPUT #1, x$
  NEXT x%
  counter% = counter% + 1
WEND
CLOSE

```

```

IF counter% <> 256 THEN
  PRINT "Falsche Datei"
  END
END IF

REM ega-neu.bin enthält 4096 Bytes = 256 * 16
REM =====

OPEN "ega-neu.asc" FOR INPUT AS #1
OPEN "ega-neu.bin" FOR BINARY AS #2

REM 8x16-Byte-Zeichen: 16 Zeile zu je 8 Bildpunkten
REM =====

puffer$ = "0123456789ABCDEF": REM Länge 16

FOR x% = 1 TO 256
  LINE INPUT #1, x$: REM Kommentarzeile
  FOR y% = 1 TO 16
    LINE INPUT #1, zeile$
    GOSUB binto hex
    MID$(puffer$, y%, 1) = byte$
  NEXT y%
  PUT #2, , puffer$
NEXT x%
CLOSE

REM Umwandlung von ega.cpi in Sanskrit-Transliteration/5.3.1989/us
REM =====

PRINT
PRINT "EGA-KON3.BAS"
PRINT "======"
PRINT
PRINT "3. Schritt: Vereinigung von EGA-ALT.CPI + EGA-NEU.BIN"
PRINT "          zu EGA-NEU.CPI"

REM Struktur
REM =====

REM EGA-ALT.CPI          EGA-NEU.BIN          EGA-NEU.CPI
REM ----->

REM 1. Header ----->
REM (65 Bytes)

REM          2. Zeichensatz ----->
REM          (4096 Bytes)

REM 3. Rest ----->
REM von alt.cpi

REM Dateien müssen schon existieren!
REM Vorher mit COPY duplizieren!

OPEN "ega-alt.cpi" FOR INPUT AS #1
l1 = LOF(1): CLOSE #1
OPEN "ega-neu.cpi" FOR INPUT AS #2
l2 = LOF(2): CLOSE #2
IF l1 <> l2 THEN
  PRINT "ega-neu.cpi muß erst dupliziert werden"
  END
END IF

```

```

REM ega-neu.bin in ega-neu.cpi hineinkopieren
REM =====

OPEN "ega-alt.cpi" FOR BINARY AS #1
OPEN "ega-neu.bin" FOR BINARY AS #2
OPEN "ega-neu.cpi" FOR BINARY AS #3

REM Header überspringen
REM -----

x$ = SPACE$(65)
GET #1, , x$
GET #3, , x$

REM Neuer Zeichensatz
REM -----

x$ = SPACE$(4096)
GET #2, , x$
PUT #3, , x$

CLOSE

END

REM 8stellige ASCII-Binär-Zeile "####...." = zeile$
REM in 1stelliges String-Byte umwandeln    = byte$

binto hex:

byte% = 0
FOR b% = 0 TO 7
  b$ = MID$(zeile$, b% + 1, 1)
  IF b$ = "#" THEN
    byte% = (byte% OR bin%(b%))
  END IF
NEXT b%
byte$ = MKI$(byte%)
byte$ = LEFT$(byte$, 1)

RETURN

```

## **CONFIG.SYS**

```

FILES=20
BUFFERS=15
COUNTRY=49,437,C:\DOS\COUNTRY.SYS
SHELL=C:\COMMAND.COM /P /E:256
INSTALL=C:\DOS\KEYB.COM GR,,C:\DOS\KEYBOARD.SYS
REM Zeilen entfernen für Normal-Keyboard
INSTALL=C:\DOS\NLSFUNC.EXE C:\DOS\COUNTRY.SYS
DEVICE=C:\DOS\DISPLAY.SYS CON:=(EGA,437,1)

```

## **AUTOEXEC.BAT**

```

@ECHO OFF
SET COMSPEC=C:\COMMAND.COM
PATH C:\;C:\DOS;
REM C:\DOS\KEYB GR,,C:\DOS\KEYBOARD.SYS statt config.sys
REM C:\DOS\NLSFUNC C:\DOS\COUNTRY.SYS statt config.sys
REM Folgendes entfernen für Normal-Keyboard
C:\DOS\MODE CON: CP PREPARE=((437) C:\DOS\EGA-SANS.CPI)
CHCP 437

```

0	5	10	15	20	25	30	35 - #
1	6	11	16	21	26	31	36 - \$
2	7	12	17	22	27	32 -	37 - %
3	8	13	18	23	28	33 - !	38 - &
4	9	14	19	24	29	34 - "	39 - '

40 - (	45 - -	50 - 2	55 - 7	60 - <	65 - A	70 - F	75 - K
41 - )	46 - .	51 - 3	56 - 8	61 - =	66 - B	71 - G	76 - L
42 - *	47 - /	52 - 4	57 - 9	62 - >	67 - C	72 - H	77 - M
43 - +	48 - 0	53 - 5	58 - :	63 - ?	68 - D	73 - I	78 - N
44 - ,	49 - 1	54 - 6	59 - ;	64 - @	69 - E	74 - J	79 - O



80 - P	85 - U	90 - Z	95 - _	100 - d	105 - i	110 - n	115 - s
81 - Q	86 - V	91 - [	96 - `	101 - e	106 - j	111 - o	116 - t
82 - R	87 - W	92 - \	97 - a	102 - f	107 - k	112 - p	117 - u
83 - S	88 - X	93 - ]	98 - b	103 - g	108 - l	113 - q	118 - v
84 - T	89 - Y	94 - ^	99 - c	104 - h	109 - m	114 - r	119 - w









## Anhang: Code Page 437 – Alter IBM-PC-DOS-Zeichensatz

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000		☺	☹	♥	♦	♣	♠	●	◼	○	◉	♂	♀	♪	♫	☼
016	▶	◀	↕	!!	¶	§	■	↕	↑	↓	→	←	└	↔	▲	▼
032		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
048	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
064	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
080	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
096	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	p	q	r	s	t	u	v	w	x	y	z	{		}	~	␣
128	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
144	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	₣	f
160	á	í	ó	ú	ñ	Ñ	²	³	¿	¬	¬	½	¼	¡	«	»
176	⋮	⋮	⋮		┆	┆	┆	π	₯	┆		₯	┆	┆	┆	┆
192	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆
208	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	■	■	■	■	■
224	α	β	Γ	Π	Σ	σ	μ	τ	Φ	Θ	Ω	δ	∞	∅	ε	∩
240	≡	±	≥	≤	∫	J	÷	≈	°	•	•	√	π	²	■	

blau = bei 437 unbelegt (NUL und Leertasten)

rot = unbelegbar unter MS-Word (Windows)

grün = theoretisch belegbar unter MS-Word

**Hinweis:** Der Bereich 0-31 (hex 00-1F) wurde früher unter DOS doppelt genutzt, erstens als unsichtbare Steuerzeichen (z.B. CR, LF, FF etc.) und zweitens als sichtbare Sonderzeichen. Ein Steuerzeichen ist auch DEL (hex 7F), das als sichtbares Zeichen ein Häuschen darstellte.

## Anhang: Bemerkungen zu Nadeldruckern

Einer der ersten Nadeldrucker (Matrixdrucker) war der EPSON MX-80 F/T Dot Matrix Printer, den ich 1981 kaufte. Die Buchstaben hatten bei diesem Drucker eine 9 x 9 Dot-Matrix, wobei zunächst nur GROSSBUCHSTABEN und später nach Kauf des Kleinschreibumrüstsatzes auch Kleinbuchstaben (mit echten Unterlängsen) und deutsche Umlaute gedruckt werden konnten. Man beachte, daß die Umlaute (ÄÖÜ), das Eszett (ß) und Paragraph-Zeichen (§) in den ASCII-Zeichensatz (unten in der Tabelle selbst hervorgehoben) hinein "gepatcht" waren.

abcdefghijklmnopqrstuvwxyz  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ

Scan des Alphabets des 9-Nadel-Druckers Epson MX-80

### A BRIEF HISTORY of APPLE

by Michael M. Scott, President

Presented at the September 12, 1978 meetings of A.P.P.L.E

Apple was started two and a half years ago by two gentlemen, Steve Jobs and Steve Wozniak who met at the "home brew" computer club at the Stanford accelerator. There, they got together and put into manufacture the Apple I, which was a single-board, black and white -basically a fancy monitor- that worked with a TV set. A year later, they were joined by three other gentlemen, that's myself (Mike Scott), Mike Markkula and Rod Holt, and formed Apple Computer, inc. and went about the business of making the Apple II.

Scan eines Textes von Michael M. Scott, der mit einem 7-Nadel-Drucker (mit Kleinbuchstaben ohne Unterlängsen) gesetzt worden war.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
032		!	"	#	\$	%	&	'	(	)	*	+	,	=	.	/
048	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
064	§	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
080	P	Q	R	S	T	U	V	W	X	Y	Z	Ä	Ö	Ü	^	_
096	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	P	q	r	s	t	u	v	w	x	y	z	ä	ö	ü	ß	

Faksimile-TrueType-Font zur Emulation eines alten 7-Nadel-Matrixdruckers

Bei den alten 7-Nadel-Matrixdruckern hatten die Kleinbuchstaben, sofern sie überhaupt vorhanden waren, keine Unterlängsen: aqpx.