



Single-Chip, Semi-Graphic-Display-Processor

Frische Farben braucht der Mensch

Schöne, vor allem aber bunte Grafik läßt sich mit einer relativ einfachen Schaltung verwirklichen, die natürlich auch an den NDR-Klein-Computer paßt.

```

0000'      cseg
0000' C3 004E'  jp start      ; dort Hauptprogramm
0020'      ef equ 20h      ; Port data
0021'      efs equ 21h     ; Port Timing
                        ; Fortbelegung, des Timing-Ports
                        ; 7 6 5 4 3 2 1 0
                        ; x x x -cs r/-w ds as -enable
0003'      initef::
0003' 3E 19      ld a,11001b   ; sperren, as,ds=0
0005' D3 21      out (efs),a   ; damit Schnittstelle bereit
0007' C9          ret
                        ; Lesen von Daten
0008' readef::
0008' 78          ld a,b       ; b=adresse -> c=data
0009' D3 20      out (ef),a   ; Adresse ausgeben
000B' 3E 04      ld a,01010b   ; AS pulsen fuer Adresseubernahme
000D' D3 21      out (efs),a   ; und cs setzen,
000F' 3E 08      ld a,01000b   ; und AS wieder low
0011' D3 21      out (efs),a   ; dann DS-Strobe
0013' 3E 0D      ld a,01101b   ; und enable frei
0015' D3 21      out (efs),a
0017' D8 20      in a,(ef)   ; Wert holen
0019' 4F          ld c,a       ; und laden
001A' 3E 19      ld a,11001b   ; as,ds=0 cs=1
001C' D3 21      out (efs),a   ; ende
001E' C9          ret
001F' writeef::
001F' 78          ld a,b       ; b=adresse c=data
0020' D3 20      out (ef),a   ; Adresse ausgeben
0022' 3E 02      ld a,00010b   ; AS pulsen fuer Adresseubernahme
0024' D3 21      out (efs),a   ; und cs setzen, r/w=0
0026' 3E 00      ld a,00000b   ; und AS wieder low
0028' D3 21      out (efs),a   ; dann DS-Strobe
002A' 79          ld a,c       ; Data ausgeben
002B' D3 20      out (ef),a   ;
002D' 3E 04      ld a,00100b   ; enable frei, DS
002F' D3 21      out (efs),a   ;
0031' 3E 19      ld a,11001b   ; as,ds=0 cs=1
0033' D3 21      out (efs),a   ; ende
0035' C9          ret
0036' waitef::
0036' F5          push af      ; warten bis letzter Befehl
0037' C5          push bc     ; ausgefuehrt wurde.
0038' wal:
0038'          ; dazu Status abfragen
0039' 06 20      ld b,20h     ; mit der Read-Funktion
003A' CD 0008'  call readef
003D' 79          ld a,c
003E' E6 80      and 80h     ; Busy Bit maskieren
0040' 20 F6      jr nz,wal    ; =0, dann fertig, sonst weiter
0042' C1          pop bc
0043' F1          pop af
0044' C9          ret
0045' writwa::
0045'          ; warten bis Befehl ausgefuehrt
0045'          ; und dann schreiben
0045' CD 0036'  call waitef   ; c=data, b=adresse
0048' C3 001F'  jp writeef
0048' start:
0048'          ; Hauptprogramm
0048' 31 0066'  ld sp,stack ; Stackpointer definieren
004E' CD 0003'  call initef ; Port auch definieren
0051' 06 2B      ld b,2Bh     ; dummy command, um
0053' 0E 00      ld c,0       ; ef9345 zu aktivieren
0055' CD 0045'  call writwa ; XDR=1, ausgeben RESET
0058' 06 21      ld b,21h     ; dann Dataregister belegen
005A' 0E 00      ld c,0h     ; CSYNC, 625 Lines/ 402/Zeile long
005C' CD 0045'  call writwa ; und danach Befehl ausloesen
005F' 06 2B      ld b,2Bh     ; CMD ausgeben
0061' 0E B1      ld c,81h     ; Aux-Register ist TSB
0063' CD 0045'  call writwa
                        ;
0066' 06 21      ld b,21h     ; wieder erst Datenregister belegen
0068' 0E 63      ld c,63h     ; Background=gelb, cursorrein
006A' CD 0045'  call writwa ; und dann Befehl dazu geben
006D' 06 2B      ld b,2Bh     ; CMD ausgeben
006F' 0E B2      ld c,82h     ; MAT-Register
0071' CD 0045'  call writwa
                        ;
0074' 06 21      ld b,21h     ; Datenregister
0076' 0E 7F      ld c,07fh    ; all enable, 40 I/Mode , long mode
0078' CD 0045'  call writwa ;
0079' 06 2B      ld b,2Bh     ; CMD

```

Bild 2: Beispielprogramme, die die Funktion verdeutlichen.

```

007D' 0E B3      ld c,83h     ; FAT-Register
007F' CD 0045'  call writwa
                        ;
0082' 06 21      ld b,21h     ; Datenregister
0084' 0E 08      ld c,8       ; start=8, Block=0
0086' CD 0045'  call writwa ; row von 8 bis 31 max.
0089' 06 2B      ld b,2Bh     ;
008B' 0E B7      ld c,87h     ; ROR Register
008D' CD 0045'  call writwa
                        ;
0090' 06 21      ld b,21h     ; Datenregister
0092' 0E A0      ld c,0a0h    ; Quadrichrome=obere Speicherhaelfte
0094' CD 0045'  call writwa ; Seigraphic=weiter Block
                        ;
0097' 06 2B      ld b,2Bh     ; und ausgeben
0099' 0E B4      ld c,84h     ;
009B' CD 0045'  call writwa ; ins DOR-Register
                        ;
009E' 0E 26      ld c,26h     ; x=0, y=8, Main pointer a D Y
00A0' 06 08      ld b,8       ; y=8, erste Zeile.
00A2' CD 0045'  call writwa
00A5' 0E 27      ld c,27h     ; Main pointer b
00A7' 06 00      ld b,0       ;
00A9' CD 0045'  call writwa ; B X
                        ;
00AC' clearseent::
00AC' 2E 00      ld l,0       ; alles loeschen
00AE' ;1         do e,32     ; std erst bei 0 beginnen
00B0' .L1:
00B0' 06 26      ld b,26h     ; c=0=Status Zeile
00B2' 4D          ld c,l       ; y=8 start rows.
00B3' CD 0045'  call writwa
00B6' 06 27      ld b,27h     ; x-Position
00B8' 0E 00      ld c,0       ; x=0
00BA' CD 0045'  call writwa
00BD' 06 21      ld b,21h     ; Wort 1 Daten laden
00BF' 0E 20      ld c,20h     ; Blank, ASCII-Code
00C1' CD 0045'  call writwa
00C4' 06 22      ld b,22h     ; Wort 2
00C6' 0E 00      ld c,0       ; keine Attribute, STD-Satz
00C8' CD 0045'  call writwa
00CB' 06 23      ld b,23h     ; Wort 3
00CD' 0E 12      ld c,12h     ; fg=rot, bk=gruen
00CF' CD 0045'  call writwa
                        ;
00D2' 16 2B      ;2         do d,40
00D4' .L2:
00D4' 06 2B      ld b,2Bh     ;
00D6' 0E 01      ld c,1       ; wrt 3, autoinc
00D8' CD 0045'  call writwa ; Zeile einschreiben.
00DB' .L1:
00DB' 15          DEC D
00DC' C2 00D4'  JP NZ,.L2
00DE' 2C          inc l
00E0' .L1:
00E0' 10          DEC E
00E1' C2 00B0'  JP NZ,.L1
                        ;
00E4' 06 26      ld b,26h     ; Statuszeile laden
00E6' 0E 00      ld c,0       ; y=0 start Statuszeile
00E8' CD 0045'  call writwa
00EA' 06 27      ld b,27h     ;
00EC' 0E 00      ld c,0       ; x=0
00EE' CD 0045'  call writwa
00F0' 06 21      ld b,21h     ; Daten laden
00F2' 0E 20      ld c,20h     ; Blank
00F4' CD 0045'  call writwa
00F6' 06 22      ld b,22h     ;
00F8' 0E 00      ld c,0       ; keine Attribute, STD-Satz
00FA' CD 0045'  call writwa
0100' 06 23      ld b,23h     ;
0102' 0E 16      ld c,16h     ; fg=rot, bk=gruen+blau
0104' CD 0045'  call writwa
0107' 21 02AF'  ld hl,hallo ; nur noch Datenwert aendern
010A' lpp:
010A' 7E          ld a,(hl)
010B' B7          or a
010C' 28 11      jr z,fin
010E' 06 21      ld b,21h     ;
0110' 4F          ld c,a       ; Parameter Data laden
0111' CD 0045'  call writwa ; und schreiben
0114' 06 2B      ld b,2Bh     ; und ablegen
0116' 0E 01      ld c,1       ; Autoinc
0118' CD 0045'  call writwa
011B' 23          inc hl
011C' C3 010A'  JP lpp     ; und neue Daten schreiben
011F' finat:
011F'          ; Ende Statuszeile schreiben

```

Eine sehr interessante und preiswerte Methode, zu einer Farbgrafik zu kommen, bietet das neue IC EF9345 von Thomson. Dieser Baustein enthält die komplette Logik eines CRT-Controllers in einem einzigen IC. Man muß nur noch Speicher und IO-Logik hinzufügen und fertig ist das CRT-Interface. Der Baustein besitzt drei Ausgänge R, G, B,

an denen direkt die Farbsignale anstehen. Zwei Möglichkeiten der Bildschirmdarstellung existieren. Die 25 Zeilen können mit 40 oder mit 80 Zeichen gefüllt werden, so daß man ihn auch als Textterminal verwenden kann. Dabei besitzt der Baustein einen eingebauten Zeichengenerator, der nicht nur den amerikanischen Satz beinhaltet, sondern

auch deutsche Umlaute und die Sonderzeichen anderer Sprachen. Ebenfalls verfügbar ist die Blockgrafik, wie sie vom Bildschirmtext her bekannt ist und auch diverse Semigrafikzeichen. Der Clou aber ist der programmierbare Zeichengenerator, wobei es bei maximalem Speicherausbau (16 KByte) mehrere geben kann. Mit dem sogenannten Quadrichrom-

Zeichengenerator lassen sich auch mehrfarbige Zeichen definieren. Da die Schaltung nicht sehr umfangreich ist, kann man sie zum Beispiel auf der IOE-Baugruppe des NDR-Klein-Computers aufbauen. Bild 1 zeigt die gesamte Schaltung. Dabei wurde gleich der maximale Speicherausbau vorgesehen, also 16 KBytes. Als Speicher wurden die Bau-

Huckepack kann die Schaltung auf den NDR-Klein-Computer geladen werden

steine 81416 von Fujitsu verwendet, die aber auch von Texas Instruments als TMS 4416 geliefert werden. Dieser Baustein ist 16 K x 4 organisiert und man braucht nur zwei davon.

Die detaillierte Funktion des EF 9345 ist deshalb kompliziert, weil er über sehr viele Optionen verfügt, so daß es einfacher ist ein Programmbeispiel zu verwenden, mit dem jeder die Funktion nachprüfen kann. Das Manual zum Baustein ist von der Herstellerfirma erhältlich.

Bild 2 zeigt das Programm. Es wurde unter CP/M geschrieben, kann aber leicht für andere Speicherbereiche umgeschrieben werden, da es keine speziellen Funktionen von Monitor oder Betriebssystem verwendet.

Der IO-Port wird auf Adresse 20H und 21H adressiert. Der Baustein selbst verfügt über einen gemultiplexten Adreß- und Datenbus. Damit dies nicht zu Problemen führt, wurden die Signale zur Steuerung über Software erzeugt. Der Port 20H ist somit der Datenport, an den sowohl Daten als auch Adressen gesendet werden können und von dem Daten gelesen werden können. Port 21H ist der Steuerport, mit dem die Signale AS, DS, R/W und CS erzeugt werden.

In dem Programm sorgen die Unterprogramme INITEF, READEF und WRITEF für die Steuerung dieser Signale. Das Unterprogramm WAITEF wartet solange, bis der Baustein für eine Befehlsaufnahme be-

reit ist und mit WRITWA wird dann auch gleich noch anschließend das Datenbyte übergeben.

Das Hauptprogramm beginnt bei der Marke START.

Das Unterprogramm INITEF sorgt dafür, daß der IO-Port die Steuersignale in einen definierten Zustand bringt. Dann wird als erstes ein „dummy command“ ausgegeben, das dafür sorgt, daß von jetzt an alle Statuswerte gültig sind.

Da man den CRT-Controller für unterschiedliche Displayformate programmieren kann, wird jetzt in einer längeren Sequenz alles definiert, was zum Betrieb nötig ist.

Unser Formatbeispiel wird 40 Zeichen pro Zeile, mit 25 Zeilen sein, wobei die oberste Zeile eine zusätzliche Statuszeile darstellt.

Jedes Zeichen auf dem Bildschirm benötigt bei der gewählten Darstellungsart drei Bytes. Das erste Byte bestimmt den Zeichencode, das zweite Byte bestimmt die Attribute, wie Blinken, Underline, Größe und welcher Zeichengenerator verwendet werden soll. Dabei kann man auch den externen Zeichengenerator angeben. Bei normalen Zeichen bestimmt das dritte Byte die Vordergrund- und Hintergrundfarbe, bei den sogenannten Quadrichrom-Zeichen wird ein Farbcode angegeben, der bestimmt, welche vier Farben in dem Zeichen vorkommen.

Im Programmteil CLEARSCREEN wird der Bildschirm gelöscht. Dabei werden alle drei Bytes in den Speicher eingeschrieben. Ein AUTOINCREMENT-Mode des EF9345 hilft beim Einschreiben und sorgt dafür, daß man die Schreibadressen nicht immer erhöhen muß. Die Speicherorganisation ist etwas eigentümlich, denn alles ist in 40-Zeichen-Rastern zu adressieren. Man gewöhnt sich aber schnell daran.

```

; Datenzeilen schreiben.
011F 06 26 ld b,26h
0121 0E 08 ld c,8 ; y=8 start Textzeile
0123 CD 0045 call writwa
0126 06 27 ld b,27h
0128 0E 00 ld c,0 ; x=0
012A CD 0045 call writwa
012D 06 21 ld b,21h ; Daten laden
012F 0E 20 ld c,20h ; Blank
0131 CD 0045 call writwa
0134 06 22 ld b,22h
0136 0E 00 ld c,0 ; keine Attribute, STD-Satz
0138 CD 0045 call writwa
0139 16 00 ld d,00h ; Vordergrundfarbe aendern
; und Inversbit bei 9.16
;1
013D 06 10 LD b,16
013F .L3:
013F C5 push bc
0140 D5 push de
;
0141 7A ld a,d
0142 F6 02 or 2 ; Hintergrund=gruen
0144 4F ld c,a
0145 06 23 ld b,23h
0147 CD 0045 call writwa
;
014A 21 02DF lpa: ld hl,data
014D 7E ld a,(hl)
014E B7 or a
014F 2B 10 jr z,final
0151 4F ld c,a
0152 06 21 ld b,21h
0154 CD 0045 call writwa
0157 06 28 ld b,28h
0159 0E 01 ld c,1 ; inc=1
015B CD 0045 call writwa
015E 23 inc hl
015F 18 EC jr lpa
0161 final:
; y increment-Befehl
0161 06 28 ld b,28h
0163 0E 80 ld c,0b0h
0165 CD 0045 call writwa
0168 06 27 ld b,27h
016A 0E 00 ld c,0 ; x auf 0 anschliessend
;
016F D1 pop de
0170 7A ld a,d
0171 C6 10 add a,10h ; Vordergrundfarbe
0173 57 ld d,a
0174 C1 pop bc
;1
0175 05 DEC B
0176 C2 013F JP NZ,.L3
;
; Spezialzeichen auch ausgeben,
; eingebauter Zeichengenerator
0179 06 26 ld b,26h
017B 0E 19 ld c,25
017D CD 0045 call writwa
0180 06 27 ld b,27h
0182 0E 00 ld c,0
0184 CD 0045 call writwa ; Koordinate eingestellt
0187 06 22 ld b,22h
0189 0E 20 ld c,20h
018B CD 0045 call writwa
;
018E 2E 00 ld 1,0 ;Startwert
;1
0190 16 06 do d,6
0192 .L4: LD d,6
;2
0192 1E 28 do e,40
0194 .L5: LD e,40
0194 06 21 ld b,21h
0196 4D ld c,l ; Daten-Code ausgeben
0197 CD 0045 call writwa
019A 7D ld a,l
019B 07 r1ca
019C 07 r1ca
019D 07 r1ca
019E E6 78 and 78h
01A0 F6 02 or 2 ; gruen als BK
01A2 4F ld c,a
01A3 06 23 ld b,23h ; Farbcode
01A5 CD 0045 call writwa
01A8 06 28 ld b,28h
01AA 0E 01 ld c,1 ; Schreiben
01AC CD 0045 call writwa
01AF 2C inc l
;2
01B0 1D enddo
01B1 C2 0194 DEC E
JP NZ,.L5
; y increment-Befehl
01B4 06 28 ld b,28h
01B6 0E 80 ld c,0b0h
01B8 CD 0045 call writwa
01BB 06 27 ld b,27h
01BD 0E 00 ld c,0
01BF CD 0045 call writwa ; x auf 0 anschliessend
;1
01C2 15 enddo
01C3 C2 0192 DEC D
JP NZ,.L4
;
; SCROLL durchfuehren
; immer schneller werdend
01C6 21 7530 ld hl,30000
01C9 22 0000 ld (merker),hl ; Warteschleife, Startwert
01CC 16 08 ld d,8 ; Scroll offset 8..31
;1
01CE 21 01F4 do hl,500
01D1 .L6: LD hl,500
01D1 06 21 ld b,21h
01D3 4A ld c,d
01D4 CD 0045 call writwa
01D7 06 28 ld b,28h
01D9 0E 87 ld c,87h
    
```

```

010B' CD 0045' call writwa
010E' B5 push de
010F' ED 5B 0000' ld de,(merker)
;2
01E3' .L7: do de
01E3' 00 nop ; Delay
;2
01E4' 1B enddo
01E5' 7A DEC DE
01E6' B3 LD A,D
01E7' C2 01E3' DR E
01E8' E5 JP NZ,.L7
01EB' 2A 0000' push hl ; immer Schneller werden
01EE' 11 0320' ld hl,(merker)
01F1' AF ld de,B00
01F2' ED 52 xor a
;2
01FA' FA 01FA' sbc hl,de
01F7' 22 0000' if p
JP M,.LB
ld (merker),hl
endif
;2
01FA' .LB:
01FA' E1 pop hl
01FB' D1 pop de
01FC' 14 inc d
;2
01FD' 7A if d=32
01FE' FE 20 LD A,d
0200' C2 0205' CP 32
0203' 16 0B JP NZ,.L9
ld d,B
endif
;2
0205' .L9:
0205' 2B enddo
0206' 7C DEC HL
0207' B5 LD A,H
0208' C2 01D1' DR L
JP NZ,.L6
; Quadrichroe Test, programmierbare Zeichen
; Zeichensatz Code 0..3, 32..127
; sind erlaubt, pro Set
0208' 06 24 ld b,24h ; Z2,D6-C2
020D' 0E 00 ld c,0
020F' CD 0045' call writwa
0212' 06 25 ld b,25h ; Z0/Z1, NT, C1/CO
0214' 0E 00 ld c,0 ; lsb-Zaehler
0216' CD 0045' call writwa
0219' 06 26 ld b,26h ; Z3
021B' 0E 40' ld c,40h ; oberere Speicherhaelfte
021D' CD 0045' call writwa
0220' fill:
0220' 21 0287' ld hl,chartab ; Tabelle Basisadresse
0223' 1E 00 ld e,0 ; char 0..3
;1
0225' 0E 04 do c,charnr ; hier max 4 Zeichen
LD c,charnr
.L10:
0227' 16 00 ld d,0 ; slice auf 0
do b,10 ; Slicennummer
LD b,10
;2
0229' 06 0A .L11:
022B' 7A ld a,d
022C' 07 r1ca
022D' 07 r1ca
022E' E6 3C and 00111100b ; Slice
0230' B3 or e ; 0,1,2,3 C1/CO
0231' 14 inc d
0232' C5 push bc
0233' 06 25 ld b,25h
0235' 4F ld c,a ; ausgeben
0236' CD 0045' call writwa ; an Pointer
0239' 06 21 ld b,21h
023B' 4E ld c,(hl)
023C' 23 inc hl
023D' CD 0045' call writwa
0240' 06 2B ld b,28h
0242' 0E 34 ld c,34h ; aux pointer, write
0244' CD 0045' call writwa
0247' C1 pop bc
;2
0248' 05 enddo
0249' C2 022B' DEC B
024C' 1C JP NZ,.L11
inc e
enddo
;1
024D' 0D DEC C
024E' C2 0227' JP NZ,.L10
;
; jetzt zeichen noch einschreiben.
0251' 06 26 ld b,26h
0253' 0E 0A ld c,10 ; y=10 start Datenreihe
0255' CD 0045' call writwa *
0258' 06 27 ld b,27h
025A' 0E 13 ld c,19 ; x=19, etwa mitte
025C' CD 0045' call writwa
025F' 06 22 ld b,22h
0261' 0E C0 ld c,110000000h ; Quadrichroe Satz 0
0263' CD 0045' call writwa
0266' 06 23 ld b,23h
0268' 0E D2 ld c,110100100h ; Farbsatz, white,cyan,blue,red
026A' CD 0045' call writwa
026D' 0E 00 ld c,0
do d,4
LD d,4
;1
026F' 16 04 .L12:
0271' 06 21 ld b,21h ; Codes ausgeben
0273' C5 push bc ; 0,1,2,3 ausgeben als Zeichen
0274' CD 0045' call writwa
0277' 06 2B ld b,28h
0279' 0E 01 ld c,1 ; Write Command
027B' CD 0045' call writwa ; mit autoincr.
027E' C1 pop bc
027F' 0C inc c
enddo
;1
0280' 15 DEC D
0281' C2 0271' JP NZ,.L12
;
; Programmende zurueck ins CP/M
0284' CD 0000 call 0 ; Texte und Daten
page

```

Wenn Sie wirklich
wissen wollen, wie ein
Computer funktioniert:
Bauen Sie ihn doch einfach selbst!



Das Einsteigerpaket:
Bausätze DM 398,-

Der NDR-Computer aus dem Fernsehen – ein Selbstbau-Computer mit unbegrenzten Möglichkeiten!

Steigen Sie klein ein mit dem NDR-Computer – schon für etwa DM 400,- können Sie sich einen funktionsfähigen Computer selbst bauen, der später zum Profi-System in verschiedenen Variationen (z.B. 16 Bit oder CP/M) ausgebaut werden kann.

ger Aktuell

Jetzt lieferbar:

Betriebssystem CP/M 2.2 für NDR 398.-
TEAC-Laufwerke FD55F, dazu passend. 698.-
FLO2 Floppy-Disc-Controller-Karte 389.-
Dynamische RAM-Karte 64 KByte od. 256 KByte
Bausatz (ohne RAMs) 129.-
64 KByte RAMs 129.-
256 KByte RAMs, billiger 445.-

Wir
stellen
aus:



Mittwoch, 17. – Mittwoch, 24. April
Hannover Messe '85
Halle 12 / 2. OG / Stand 2321

Lernen Sie mit dem NDR-Computer: Durch den Selbstbau lernen Sie wirklich, wie ein Computer funktioniert. Sie lernen auch bei der Programmierung: Beginnend bei der Maschinenprogrammierung im Einsteigerpaket bis zu allen wichtigen Programmiersprachen beim späteren Ausbau.

Natürlich gibt es auch alle Baugruppen fertig aufgebaut und geprüft.

Sie entscheiden sich für einen Computer, der nie veralten wird! Der NDR-Computer besteht aus kleinen Einheiten, die leicht erweitert oder ausgetauscht werden können. Damit sind Sie immer mit vorne dran!

Der NDR hat sich für unseren Computer entschieden – tun Sie es auch!

Sie investieren in Ihre Zukunft – fordern Sie heute noch unsere ausführliche, kostenlose Info + Probeexemplar unserer Kundenzeitung an (bitte DM 1,40 Briefmarken für Rückporto beifügen).

Graf Elektronik Systeme GmbH

8960 Kempten · Telefon (0831) 6211

Filiale Hamburg:
Ehrenbergstr. 56 · 2000 Hamburg 50 (Altona)
Telefon (040) 38 8151

Filiale München:
Georgenstr. 61 · 8000 München 40 (Schwabing)
Telefon (089) 2 7158 58

Schweiz: SYSTECH · Starenstr. 21 · CH-4106 Therwil



```

0287'      chartab:      ; Zeichen fuer prog. Generator,
                        ; quadrichrome Mode
0287' 00      defb 00000000b
0288' FC      defb 1111100b
0289' FC      defb 1111100b
028A' 5C      defb 0101100b
028B' FC      defb 1111100b
028C' FC      defb 1111100b
028D' 5C      defb 0101100b
028E' FC      defb 1111100b
028F' FC      defb 1111100b
0290' 00      defb 00000000b
                        ;
0291' 00      defb 00000000b
0292' FF      defb 11111111b
0293' 57      defb 01010111b
0294' F5      defb 11110101b
0295' F6      defb 11110110b
0296' F4      defb 11110110b
0297' F5      defb 11110101b
0298' 57      defb 01010111b
0299' FF      defb 11111111b
029A' 00      defb 00000000b
                        ;
029B' 00      defb 00000000b
029C' FF      defb 11111111b
029D' FD      defb 11111011b
029E' FD      defb 11111011b
029F' 55      defb 01010101b
02A0' FD      defb 11111011b
02A1' FD      defb 11111011b
02A2' FD      defb 11111011b
02A3' FF      defb 11111111b
02A4' 00      defb 00000000b
                        ;
02A5' 00      defb 00000000b
02A6' 3F      defb 00111111b
02A7' 3F      defb 00111111b
02A8' 3F      defb 00111111b
02A9' 35      defb 00110101b
02AA' 3F      defb 00111111b
02AB' 3F      defb 00111111b
02AC' 3F      defb 00111111b
02AD' 3F      defb 00111111b

02AE' 00      defb 00000000b
0004      charnr equ 4      ; Anzahl Zeichen

02AF'      haloc:
02AF' 48 61 6C 6D      defb 'Hallo Test EF9345 - Statuszeile',0
02B3' 6F 20 54 65
02B7' 73 74 20 45
02BB' 46 39 33 34
02BF' 35 20 2D 2D
02C3' 53 74 61 74
02C7' 75 73 7A 65
02CB' 69 6C 65 00
02CF'      data:
02CF' 44 65 72 20      defb 'Der EF9345 kann 8 Farben darstellen.',0
02D3' 45 46 39 33
02D7' 34 35 2D 68
02DB' 61 6E 6E 2D
02DF' 3B 2D 46 61
02E3' 72 62 65 6E
02E7' 2D 64 61 72
02EB' 73 74 65 6C
02EF' 6C 65 6E 2E
02F3' 00

                        ; RAM Speicher beginnt hier
02F4'      dseg
0000' 0000      merker: defb 0
0002'          defs 100
0066'          stack:
                        end start

Macros:

Symbols:
00B0' .L1      0227' .L10      022E' .L11
0271' .L12      00B4' .L2       013F' .L3
0192' .L4       0194' .L5       01D1' .L6
01E3' .L7       01FA' .L6       0205' .L9
0004' CHARNR   0287' CHARTAB  00AC' CLEARGREEN
02CF' DATA    0020' EF       0021' EFS
0220' FILL     011F' FINA    0161' FINAL
02AF' HALLO    0003' INITEF  014D' LPA
010A' LPP      0000' MERKER  000E' REDEF
0066' STACK    004B' START   0038' WA1
0036' WAITEF   001F' WRITEEF  0045' WRITWA

No fatal error(s)

```

Alle drei Bytes werden automatisch an die richtigen Stellen im Speicher geschrieben, wenn man die Datenwerte in die Daten-

kann bis zu 8 K lang sein. Hier werden aber nur vier Zeichen definiert, die einen Platz von nur 40 Bytes belegen. Diese vier Zeichen

werden dann auch in dem Bildwiederholtspeicher eingeschrieben. Anwendungen der Baugruppe sind neben BTX

auch Textverarbeitung, grafische Terminals, und natürlich können auch Spiele damit programmiert werden. *Rolf-Dieter Klein*

Bild 3:
Der Zeichensatz des EF9345.

Umlaute normale und BTX-Blockgrafik sowie der Quadrichrom-Zeichengenerator – perfekte Sache

register des Prozessors geladen hat. Nach diesem Löschen folgt das Einschreiben von Texten und Zeichen auf die gleiche Weise. Danach wird der Bildschirm in verschiedenen Geschwindigkeiten gescrollt. Dabei geschieht das Scrollen durch Verändern eines Registers des EF9345. Anschließend wird der programmierbare Zeichengenerator für Quadrichrom-Zeichen definiert. Dieser Zeichengenerator

