

[illegible]

```

<07B6> 330 DATA D301C300007EFE4E2086CDE901C300
<0827> 340 DATA 00CD5AFC51BB1115BCCD3902CBBCCB
<07F1> 350 DATA B411ECBDCD2802C91100B9CD39027C
<0783> 360 DATA FE5E115002280311E002010100CD57
<07DD> 370 DATA FC217089019000CD4BFC21708911EC
<0828> 380 DATA BDCD2802CD5AFC11BC380E21B48928
<07F4> 390 DATA 0321838911D4BDCD2802C9C5D5224B
<07EF> 400 DATA 02010100CD57FCEB114B02180BC5D5
<07EE> 410 DATA 010001CD57FC214B02010200CD4BFC
<07F9> 420 DATA 210000D1C1C9CD11BC21B48928022E
<0839> 430 DATA 8322D4BDDC51BBC314BC06004D505C
<0823> 440 DATA 602929092929292919ED4BC9B109EB
<0836> 450 DATA 26106F2929293A90B24F3ACBB182E6
<0828> 460 DATA C70608577EA9127AC6082310F6C906
<0817> 470 DATA 004D29505C6029092929292919ED4B
<0868> 480 DATA C9B109EB26106F2929293ACBB182E6
<089F> 490 DATA C7060857E5C57E21ED8977ED6F2A8F
<085B> 500 DATA B2472FA44F78A5B112133E00472FA4
<0878> 510 DATA 4F78A5B112C1E1231B7AC60810D7C9
<0790> 520 DATA CD11BC21B48928022E8322D4BDDC51
<075F> 530 DATA BBC314BC06004D505C602929092929
<0794> 540 DATA 292919ED4BC4B709EB26106F292929
<07B8> 550 DATA 3A30B74F3AC6B782E6C70608577EA9
<0769> 560 DATA 127AC6082310F6C906004D29505C60
<079F> 570 DATA 29092929292919ED4BC4B709EB2610
<07C5> 580 DATA 6F2929293AC6B782E6C7060857E5C5
<0812> 590 DATA 7E21ED8977ED6F2A2FB7472FA44F78
<07B4> 600 DATA A5B112133E00472FA44F78A5B112C1
<07E0> 610 DATA E1231B7AC60810D7C91A1A1A1A1A

```

Sourcecode

ARNOR Z80 ASSEMBLER version 1.15

Page 001

```

00003
00004
00005
00006
00007
00008
00009
00010
00011
00012
00013
00014
00015
00016 0100 (0100)
00017 0100
00018
00019
00020
00021
00022 0100 (0000)
00023 0100 (0005)
00024
00025 0100 (0970)
00026
00027 0100 (0B51)
00028 0100 (0C11)
00029 0100 (0C14)
00030 0100 (0DEB)
00031
00032 0100 (FC4B)
00033 0100 (FC57)
00034 0100 (FC5A)
00035
00036 0100 1B 1E
00037
00038
00039
00040
00041 0102 (001E)
00042 0120 (0120)
00043
00044
00045
00046
00047 0120 31 20 01
00048
00049 0123 0E 0C
00050 0125 C0 05 00
00051 0128 70
00052 0129 E6 F0
00053 012B 11 7A 01
00054 012E FE 30
00055 0130 20 21
00056
00057 0132 21 90 00
00058 0135 7E
00059 0136 A7
00060 0137 2B 67
00061
00062 0139 47
00063 013A 23
00064 013B 7E
00065 013C FE 20
00066 013E 20 02
00067 0140 10 FB
00068
00069 0142 11 50 01
00070 0145 FE 4F
00071 0147 20 0A
00072 0149 23
00073 014A 70
00074 014B FE 02
00075 014D 2B 79
00076 014F FE 03
00077 0151 20 64
00078
00079
00080
00081
00082 0153 0E 09
00083 0155 C0 05 00
00084 015B C3 00 00

;
; ***** FAST.COM *****
;
; FAST-Routine Mode 1 und 2
; fur CP/M Plus
;
; erstellt von H. Dobbertin
; am 09.03.1988
;
;
; org #0100
; write "FAST.COM"
;
; ----->>> Labeldefinitionen <<<-----
;
boot equ #0000
bdos equ #0005
;
fastmem equ #0970 ;Platz fur FAST-Routine
;
fastoff equ #0b51 ;TXT_RESET
getmode equ #0c11 ;SCR_GET_MODE
modeclr equ #0c14 ;SCR_MODE_CLEAR
newmode equ #0deb ;INDIRECTION_MODE_CLEAR
;
move equ #fc4b ;BIOS 25
xmove equ #fc57 ;BIOS 29
callsys equ #fc5a ;BIOS 30
;
jr start
;
; ----->>> Lokaler Stack fur FAST.COM <<<-----
;
defb 30
stack equ $
;
; ----->>> Programmstart <<<-----
;
start ld sp,stack ;Lokalen Stack anlegen
;
ld c,12 ;CP/M Version ermitteln
call bdos
ld a,l
and %11110000
ld de,notcpm3 ;Zeiger auf Fehlermeldung
cp #30 ;ist's CP/M Plus ?
jr nz,prt_err ;Nein, dann springe
;
ld hl,#0000 ;Zeiger auf Default-FCB
ld a,(hl) ;Parameter-Länge -> 'A'
and a ;Mit Parameter angerufen ?
jr z,toggle ;Nein, dann zu TOGGLE
;
skpspc ld b,a
inc hl ;Führende Leerzeichen
ld a,(hl) ; vor dem Parameter
cp "" ; überlesen
jr nz,tst_par
djnz skpspc
;
tst_par ld de,bad_par ;Zeiger auf Fehlermeldung
cp "0" ;Erstes Zeichen ein '0' ?
jr nz,prt_err ;Nein, dann Fehlermeldung
inc hl ;Parameterzeiger erhöhen
ld a,b ;Parameter-Länge -> 'A'
cp 2 ;Zwei Zeichen lang ?
jr z,tstfast ;Ja, dann auf '0N' testen
cp 3 ;Oder drei Zeichen ?
jr z,tstslow ;Nein, dann Fehlermeldung
;
; ----->>> Fehlermeldung ausgeben <<<-----
;
prt_err ld c,9 ;Textstring ausgeben
call bdos
jp boot ;CP/M Warestart

```

```

00085
00086 0158 07 0A 0D bad_par defb #07,#0a,#0d
00087 015E 45 72 72 6F defm "Error: Invalid parameter,"
      0162 72 3A 20 49
      0166 6E 76 61 6C
      016A 69 64 20 70
      016E 61 72 61 6D
      0172 65 74 65 72
      0176 2E
00088 0177 0A 0D 24 defb #0a,#0d,"$"
00089
00090 017A 07 0A 0D notcpm3 defb #07,#0a,#0d
00091 017D 45 72 72 6F defm "Error: Requires CPM 3 or higher,"
      0181 72 3A 20 52
      0185 65 71 75 69
      0189 72 65 73 20
      018D 43 50 40 20
      0191 33 20 6F 72
      0195 20 6B 69 67
      0199 6B 65 72 2E
00092 019D 0A 0D 24 defb #0a,#0d,"$"
00093
00094
00095 ; ----- >>> CP/M+ FAST-Routine an/abschalten <<< ---
00096 ;
00097 toggle ld de,#b0d4
00098 call rdbank0 ;LD HL,(#B0D4) aus Bank0
00099 push hl
00100 call slow
00101 pop hl ;Prüfen, ob die
00102 ld a,h ; FAST-Routine
00103 cp fastmem/256 ; vorher aktiv war
00104 jp z,boot ;Ja, dann CP/M Warmstart
00105 call fast
00106 jp boot ;CP/M Warmstart
00107
00108
00109 ; ----- >>> Teste Parameter auf 'OFF' <<< -----
00110 ;
00111 tstslow ld a,(hl) ;Zweites & drittes Zeichen
00112 cp "F" ; des Parameters auf 'F'
00113 jr nz,pnt_err ; prüfen
00114 inc hl
00115 ld a,(hl)
00116 cp "F"
00117 jr nz,pnt_err
00118
00119 call slow ;FAST-Routine abschalten
00120 jp boot ;CP/M Warmstart
00121
00122
00123 ; ----- >>> Teste Parameter auf 'ON' <<< -----
00124 ;
00125 tstfast ld a,(hl) ;Das zweite Zeichen des
00126 cp "N" ; Parameters muß 'N' sein
00127 jr nz,pnt_err
00128
00129 call fast ;FAST-Routine einschalten
00130 jp boot ;CP/M Warmstart
00131
00132
00133 ; ----- >>> CP/M+ FAST-Routine deaktivieren <<< ---
00134 ;
00135 slow call callsys ;Indirection #B0D3
00136 defw fastoff ; restaurieren
00137
00138 ld de,modeclr+1
00139 call rdbank0 ;LD HL,(#BC15) aus Bank0
00140 res 7,h ;Evtl. ROM-Select-Bits
00141 res 6,h ; löschen
00142 ld de,newmode+1 ;Indirection #B0EB
00143 call wrbank0 ; restaurieren
00144 ret
00145
00146
00147 ; ----- >>> CP/M+ FAST-Routine aktivieren <<< -----
00148 ;
00149 fast ld de,#b900
00150 call rdbank0 ;LD HL,(#B900) aus Bank0
00151 ld a,h ;Rechnerabhängiges Byte
00152 cp #5e ; holen und prüfen
00153
00154 ld de,fast464 ;CPC 464 FAST-Routine
00155 jr z,cpc_464 ;Springe wenn CPC 464
00156 ld de,fast664 ;CPC 664/6128 FAST-Routine
00157 cpc_464 ld bc,#0001 ;Ziel=Bank0, Quelle=Bank1
00158 call xmove
00159 ld hl,fastmem ;Zieladresse -> 'HL'
00160 ld bc,fastlen ;Länge -> 'BC'
00161 call move ;FAST-Routine -> Bank0
00162
00163 ld hl,fastmem ;Indirection bei #B0EB
00164 ld de,newmode+1 ; noch auf die FAST-
00165 call wrbank0 ; Routine umlenken
00166
00167 call callsys ;Den aktuellen
00168 defw getmode ; Mode holen
00169 jr c,no_fast ;Springe wenn MODE 0
00170 ld hl,fastmem+fast4_1-fast464
00171 jr z,fast_on ;Springe wenn MODE 1
00172 ld hl,fastmem+fast4_2-fast464
00173 fast_on ld de,#b0d4 ;Indirection bei #B0D3
00174 call wrbank0 ; noch patchen
00175 no_fast ret
00176
00177
00178 ; ----- >>> LD (DE),HL Schreiben in Bank0 <<< -----
00179 ;
00180 wrbank0 push bc
00181 push de
00182 ld (selfmod+1),hl
00183 ld bc,#0001 ;Ziel=Bank0, Quelle=Bank1
00184 call xmove
00185 ex de,hl ;Ziel -> 'HL'
00186 ld de,selfmod+1 ;Quelle -> 'DE'
00187 jr move_2 ;2 Byte schreiben
00188
00189
00190 ; ----- >>> LD HL,(DE) Lesen aus Bank0 <<< -----
00191 ;
00192 rdbank0 push bc
00193 push de
00194 ld bc,#0100 ;Ziel=Bank1, Quelle=Bank0
00195 call xmove
00196 ld hl,selfmod+1
00197 move_2 ld bc,#0002 ;2 Byte lesen/schreiben
00198 call move
00199 selfmod ld hl,$-$ ;Ergebnis -> 'HL'
00200 pop de

```



```

00201 024E C1          pop    bc
00202 024F C9          ret
00203
00204
00205 ; ----- >>> FAST-Routinen für CPC 464 <<< -----
00206
00207 0250 CD 11 BC      fast464 call    getmode      ;Aktuellen Mode holen
00208 0253 21 84 89      id      hl,fastmem+fast4_1-fast464
00209 0256 28 02        jr      z,newout4    ;Springe wenn MODE 1
00210 0258 2E 83        id      hl,fastmem+fast4_2-fast464 mod 256
00211 025A 22 04 BD      newout4 id      (Hb0d4),hl
00212 025D DC 51 88      call    c,fastoff    ;CALL wenn MODE 0
00213 0260 C3 14 BC      id      modeclr
00214
00215
00216 ; ----- >>> FAST-Routine Mode 2 CPC 464 <<< -----
00217
00218 ; Unter CP/M wird der ab H0DD3 stehende Indirection
00219 ; auf diese Routine umgelenkt, wenn das Programm
00220 ; FAST.COM gestartet wird.
00221
00222 0263 06 00      fast4_2 id      b,0
00223 0265 40          id      c,1          ;Zeile -> 'BC'
00224 0266 50          id      d,b
00225 0267 5C          id      e,h          ;Spalte -> 'DE'
00226 0268 60          id      h,b          ;Zeile -> 'HL'
00227 0269 29          add     hl,hl
00228 026A 29          add     hl,hl
00229 026B 09          add     hl,bc
00230 026C 29          add     hl,hl
00231 026D 29          add     hl,hl
00232 026E 29          add     hl,hl          ;In HL steht jetzt
00233 026F 29          add     hl,hl          ; Zeile + 80
00234 0270 19          add     hl,de          ;Spalte noch addieren
00235
00236 0271 ED 4B C9 B1      id      bc,(Hb1c9)    ;Zu dieser Adresse den
00237 0275 09          add     hl,bc          ; Screen-Offset addieren
00238 0276 EB          ex      de,hl          ;Screen-Adr. -> 'HL'
00239
00240 0277 26 10          id      h,#10        ;Start ROM-Zeichensatz / B
00241 0279 6F          id      l,a          ;Zeichen -> 'L'
00242 027A 29          add     hl,hl          ; # 2
00243 027B 29          add     hl,hl          ; # 4
00244 027C 29          add     hl,hl          ; # 8
00245
00246 027D 3A 90 B2      id      a,(Hb290)
00247 0280 4F          id      c,a          ;Invers-Maske -> 'C'
00248
00249 0281 3A CB B1      id      a,(Hb1cb)    ;Screen-Start -> 'A'
00250 0284 B2          add     a,d
00251 0285 E6 C7          and     %11000111
00252
00253
00254 ; HL zeigt jetzt auf die entsprechende Stelle im
00255 ; Zeichensatz, 'AE' zeigt auf die erste Bildschirm-
00256 ; speicheradresse und in Register 'C' steht die
00257 ; Invers-Maske ( H00 oder HFF ).
00258
00259 0287 06 00      loop_1 id      b,B          ;B Bytes pro Zeichen
00260 0289 57          id      d,a          ;Screenstart-High -> 'D'
00261 028A 7E          id      a,(hl)        ;Ein Byte holen
00262 028B A9          or      c
00263 028C 12          id      (de),a        ;und auf den Bildschirm
00264 028D 7A          id      a,d          ; 'HL' auf die nächste
00265 028E C6 08      add     a,H08          ; Zeile erhöhen (+ H0800)
00266 0290 23          inc     hl
00267 0291 10 F6      djnz    loop_1        ;B Zeilen pro Zeichen
00268
00269 0293 C9          ret
00270
00271
00272 ; ----- >>> FAST-Routine Mode 1 CPC 464 <<< -----
00273
00274 ; Unter CP/M wird der ab H0DD3 stehende Indirection
00275 ; auf diese Routine umgelenkt, wenn das Programm
00276 ; FAST.COM gestartet wird.
00277
00278 0294 06 00      fast4_1 id      b,0
00279 0296 4D          id      c,1          ;Zeile -> 'BC'
00280 0297 29          add     hl,hl
00281 0298 50          id      d,b
00282 0299 5C          id      e,h          ;Spalte # 2 -> 'DE'
00283 029A 60          id      h,b          ;Zeile # 2 -> 'HL'
00284 029B 29          add     hl,hl
00285 029C 09          add     hl,bc
00286 029D 29          add     hl,hl
00287 029E 29          add     hl,hl
00288 029F 29          add     hl,hl          ;In HL steht jetzt
00289 02A0 29          add     hl,hl          ; Zeile + 80
00290 02A1 19          add     hl,de          ;Spalte # 2 addieren
00291
00292 02A2 ED 4B C9 B1      id      bc,(Hb1c9)    ;Zu dieser Adresse den
00293 02A6 09          add     hl,bc          ; Screen-Offset addieren
00294 02A7 EB          ex      de,hl          ;Screen-Adr. -> 'DE'
00295
00296 02AB 26 10          id      h,#10        ;Start ROM-Zeichensatz / B
00297 02AC 6F          id      l,a          ;Zeichen -> 'HL'
00298 02AD 29          add     hl,hl          ; # 2
00299 02AE 29          add     hl,hl          ; # 4
00300 02AF 29          add     hl,hl          ; # 8
00301
00302 02AE 3A CB B1      id      a,(Hb1cb)    ;Screen-Start -> 'A'
00303 02B1 B2          add     a,d
00304 02B2 E6 C7          and     %11000111    ;Screenstart-High -> 'A'
00305
00306
00307 ; HL zeigt jetzt auf die entsprechende Stelle im
00308 ; Zeichensatz und 'AE' zeigt auf die erste Bild-
00309 ; schirmspeicheradresse.
00310
00311 02B4 06 00      loop_2 id      b,B          ;Bytes/Zeichen -> 'B'
00312 02B6 57          id      d,a          ;Screenstart-High -> 'D'
00313 02B7 E5          push    hl          ;Zeichensatz-Zeiger merken
00314 02B8 C5          push    bc
00315 02B9 7E          id      a,(hl)        ;Zeichensatz-Byte holen
00316 02BA 21 ED 89      id      hl,fastmem+patch1+1-fast464
00317 02BD 77          id      (hl),a        ; und in 2 Hälften teilen
00318 02BE ED 6F          rld          ; (2. Hälfte -> PATCH1+1 )
00319 02C0 2A BF B2      id      hl,(Hb2Bf)    ;PEN & PAPER holen
00320 02C3 47          id      b,a          ;l.Matrix in 'B' merken
00321 02C4 2F          cpl
00322 02C5 A4          and     h          ;PAPER-Bits setzen
00323 02C6 4F          id      c,a          ;PAPER-Maske in 'C' merken
00324 02C7 7B          id      a,b          ;l.Matrix wieder -> 'A'
00325 02C8 A5          and     l          ;PEN-Bits setzen und mit
00326 02C9 B1          or      c          ; PAPER-Bits verknüpfen
00327 02CA 12          id      (de),a        ;l.Byte auf den Bildschirm
00328 02CB 13          inc     de
00329 02CC 3E 00      patch1 id      a,$-5

```

```

00330 02CE 47      ld b,a      ;Z.Matrix in B merken
00331 02CF 2F      cpl      ;PAPER-Bits setzen
00332 0200 A4      and h      ;PAPER-Maske in C merken
00333 0201 4F      ld c,a      ;Z.Matrix wieder -> A
00334 0202 70      ld a,b      ;PEN-Bits setzen und mit
00335 0203 A5      and l      ; PAPER-Bits verknupfen
00336 0204 B1      or c      ;Z.Byte auf den Bildschirm
00337 0205 12      ld (de),a  ;Schleifenzahler -> B
00338 0206 C1      pop bc     ;Zeichensatz-Zeiger wieder
00339 0207 E1      pop hl     ; holen und erhoehen
00340 0208 23      inc hl
00341 0209 10      dec de
00342 020A 7A      ld a,d      ; DE auf die nachste
00343 020B C6 08   add a,#08   ; Zeile erhoehen (+ #0800)
00344 020D 10 D7   djnz loop_2 ;B Zeilen pro Zeichen
00345
00346 020F C9      ret
00347
00348 02E0 (0090)   fastlen equ $-fast664
00349
00350
00351 ; ----- >>> FAST-Routinen fur CPC 664/6128 <<< ----
00352 ;
00353 02E0 CD 11 BC   fast664 call getmode ;Aktuellen Mode holen
00354 02E3 21 B4 B9   ld hl,fastmem+fast6_1-fast664
00355 02E6 20 02      jr z,newout6 ;Springe wenn MODE 1
00356 02E8 2E 03      ld l,fastmem+fast6_2-fast664 mod 256
00357 02EA 22 D4 BD   newout6 ld (#b0d4),hl
00358 02ED DC 51 B8   call c,fastoff ;CALL wenn MODE 0
00359 02F0 C3 14 BC   jp modeclr
00360
00361 ;
00362 ; ----- >>> FAST-Routine Mode 2 CPC 664/6128 <<< -
00363 ;
00364 ; Unter CP/M wird der ab #BDD3 stehende Indirection
00365 ; auf diese Routine umgelenkt, wenn das Programm
00366 ; FAST.COM gestartet wird.
00367 ;
00368 02F3 06 00      fast6_2 ld b,0
00369 02F5 40      ld c,l      ;Zeile -> BC
00370 02F6 50      ld d,b
00371 02F7 5C      ld e,h      ;Spalte -> DE
00372 02F8 60      ld h,b      ;Zeile -> HL
00373 02F9 29      add hl,hl
00374 02FA 29      add hl,hl
00375 02FB 09      add hl,bc
00376 02FC 29      add hl,hl
00377 02FD 29      add hl,hl
00378 02FE 29      add hl,hl ;In HL steht jetzt
00379 02FF 29      add hl,hl ; Zeile # 00
00380 0300 19      add hl,de ;Spalte noch addieren
00381 ;
00382 0301 ED 40 C4 B7 ; ld bc,(#b7c4) ;Zu dieser Adresse den
00383 0305 09      add hl,bc ; Screen-Offset addieren
00384 0306 EB      ex de,hl ;Screen-Adr. -> HL
00385 ;
00386 0307 26 10      ld h,#10 ;Start ROM-Zeichensatz / B
00387 0309 6F      ld l,a ;Zeichen -> L
00388 030A 29      add hl,hl ; # 2
00389 030B 29      add hl,hl ; # 4
00390 030C 29      add hl,hl ; # 8
00391 ;
00392 030D 3A 30 B7   ; ld a,(#b730)
00393 0310 4F      ld c,a ;Invers-Maske -> C
00394 ;
00395 0311 3A C6 B7   ; ld a,(#b7c6) ;Screen-Start -> A
00396 0314 02      add a,d
00397 0315 E6 C7     and %11000111
00398 ;
00399 ;
00400 ; HL zeigt jetzt auf die entsprechende Stelle im
00401 ; Zeichensatz, AE zeigt auf die erste Bildschirm-
00402 ; Speicheradresse und in Register C steht die
00403 ; Invers-Maske ( #00 oder #FF ).
00404 ;
00405 0317 06 08      loop_3 ld b,B ;B Bytes pro Zeichen
00406 0319 57      ld d,a ;Screenstart-High -> D
00407 031A 7E      ld a,(hl) ;Ein Byte holen
00408 031B A9      xor c
00409 031C 12      ld (de),a ;und auf den Bildschirm
00410 031D 7A      ld a,d ;HL auf die nachste
00411 031E C6 08   add a,#08 ; Zeile erhoehen (+ #0800)
00412 0320 23      inc hl
00413 0321 10 F6   djnz loop_3 ;B Zeilen pro Zeichen
00414 ;
00415 0323 C9      ret
00416
00417 ;
00418 ; ----- >>> FAST-Routine Mode 1 CPC 664/6128 <<< -
00419 ;
00420 ; Unter CP/M wird der ab #BDD3 stehende Indirection
00421 ; auf diese Routine umgelenkt, wenn das Programm
00422 ; FAST.COM gestartet wird.
00423 ;
00424 0324 06 00      fast6_1 ld b,0
00425 0326 40      ld c,l      ;Zeile -> BC
00426 0327 29      add hl,hl
00427 0328 50      ld d,b
00428 0329 5C      ld e,h      ;Spalte # 2 -> DE
00429 032A 60      ld h,b      ;Zeile # 2 -> HL
00430 032B 29      add hl,hl
00431 032C 09      add hl,bc
00432 032D 29      add hl,hl
00433 032E 29      add hl,hl
00434 032F 29      add hl,hl ;In HL steht jetzt
00435 0330 29      add hl,hl ; Zeile # 00
00436 0331 19      add hl,de ;Spalte # 2 addieren
00437 ;
00438 0332 ED 40 C4 B7 ; ld bc,(#b7c4) ;Zu dieser Adresse den
00439 0336 09      add hl,bc ; Screen-Offset addieren
00440 0337 EB      ex de,hl ;Screen-Adr. -> DE
00441 ;
00442 0338 26 10      ld h,#10 ;Start ROM-Zeichensatz / B
00443 033A 6F      ld l,a ;Zeichen -> L
00444 033B 29      add hl,hl ; # 2
00445 033C 29      add hl,hl ; # 4
00446 033D 29      add hl,hl ; # 8
00447 ;
00448 033E 3A C6 B7   ; ld a,(#b7c6) ;Screen-Start -> A
00449 0341 02      add a,d
00450 0342 E6 C7     and %11000111 ;Screenstart-High -> A
00451 ;
00452 ;
00453 ; HL zeigt jetzt auf die entsprechende Stelle im
00454 ; Zeichensatz und AE zeigt auf die erste Bild-
00455 ; schirmspeicheradresse.
00456 ;
00457 0344 06 08      loop_4 ld b,B ;Bytes/Zeichen -> B
00458 0346 57      ld d,a ;Screenstart-High -> D

```

00459	0347	E5		push	hl	;Zeichensatz-Zeiger merken
00460	0348	C9		push	bc	
00461	0349	7E		ld	a,(hl)	;Zeichensatz-Byte holen
00462	034A	21	ED B9	ld	hl,fastmem+patch2+1-fast664	
00463	034D	77		ld	(hl),a	; und in 2 Hälften teilen
00464	034E	ED	6F	rld		; (2. Hälfte -> PATCH2+1)
00465	0350	2A	2F B7	ld	hl,(\$b72f)	;PEN & PAPER holen
00466	0353	47		ld	b,a	;1.Matrix in 'B' merken
00467	0354	2F		cpl		
00468	0355	A4		and	h	;PAPER-Bits setzen
00469	0356	4F		ld	c,a	;PAPER-Maske in 'C' merken
00470	0357	78		ld	a,b	;1.Matrix wieder -> 'A'
00471	0358	A5		and	l	;PEN-Bits setzen und mit
00472	0359	B1		or	c	; PAPER-Bits verknüpfen
00473	035A	12		ld	(de),a	;1.Byte auf den Bildschirm
00474	035B	13		inc	de	
00475	035C	3E	00	ld	a,\$-5	
00476	035E	47		ld	b,a	;2.Matrix in 'B' merken
00477	035F	2F		cpl		
00478	0360	A4		and	h	;PAPER-Bits setzen
00479	0361	4F		ld	c,a	;PAPER-Maske in 'C' merken
00480	0362	78		ld	a,b	;2.Matrix wieder -> 'A'
00481	0363	A5		and	l	;PEN-Bits setzen und mit
00482	0364	B1		or	c	; PAPER-Bits verknüpfen
00483	0365	12		ld	(de),a	;2.Byte auf den Bildschirm
00484	0366	C1		pop	bc	;Schleifenzähler -> 'B'
00485	0367	E1		pop	hl	;Zeichensatz-Zeiger wieder
00486	0368	23		inc	hl	; holen und erhöhen
00487	0369	18		dec	de	
00488	036A	7A		ld	a,d	; 'DE' auf die nächste
00489	036B	C6	08	add	a,\$08	; Zeile erhöhen (+ \$0800)
00490	036D	10	D7	djnz	loop_4	; 8 Zeilen pro Zeichen
00491						
00492	036F	C9		ret		