

HEWLET-PACKARD-DRUCKER-EMULATOR Version 1.2a

(c) 06.06.1989 by Hardy Lau

```
code      SEGMENT PARA
          ASSUME CS:code,DS:daten,SS:stapel

main      ;
          PROC NEAR
          MOV AX,SEG daten      ; INIT DS.
          MOV DS,AX
          ;
          MOV AX,700H           ; BILDSCHIRM LÖSCHEN
          XOR CX,CX
          MOV DX,184FH
          MOV BH,7
          INT 10H
          ;
          MOV AH,2              ; COURSOR HOME
          XOR DX,DX
          MOV BH,0
          INT 10H
          ;
          MOV AH,9              ; PRINT ANZ1
          LEA DX,anz1
          INT 21H
          ;
          CALL geschwtest       ; RECHNERGEWSCHWINDIGKEIT TESTETN
          ;
einl:      CALL lesezeichen     ; ZEICHEN EMPFANGEN
          ;
          CALL druck
          ;
          MOV AH,9              ; PRINT ANZ4
          LEA DX,anz4
          INT 21H
          ;
warzei:    MOV AH,0             ; ZEICHEN EINLESEN
          INT 16H
          CMP AL,"N"
          JE nwei
          CMP AL,"n"
          JE nwei
          CMP AL,"J"
          JE einl
          CMP AL,"j"
          JE einl
          JMP warzei
          ;
nwei:      MOV AH,9             ; PRINT ANZ9
          LEA DX,anz9
          INT 21H
          ;
          MOV AX,4C00H          ; PROGRAMM BEENDEN
          INT 21H
          ;
main      ENDP
          ;
geschwtest PROC NEAR           ; SCHNITTSTELLE KLARMACHEN
          ;
          MOV CX,30             ; WARTEN BIS LAUFWERKE AUSGESCHALTET
          MOV DX,0
del11:    ;
```

```

del12:      DEC DX
            JNE del12
            DEC CX
            JNE del11
            ;
            IN AL,21H                ;TIMER-INTERRUPT SPERREN
            OR AL,1
            OUT 21H,AL
            ;
            MOV AH,9                 ;PRINT ANZ2
            LEA DX,anz2
            INT 21H
            ;
sch_taste:  MOV AH,0                 ;AUF TASTE WARTEN
            INT 16H
            CMP AL,"1"
            JE comm1
            CMP AL,"2"
            JE comm2
            JMP sch_taste           ;FALSCHE TASTE NOCHMAL
comm1:      MOV AX,3F8H
            MOV basisport,AX        ;COMM1 AUSGEWÄHLT
            JMP doinit
comm2:      MOV AX,2F8H              ;COMM2 AUSGEWÄHLT
            MOV basisport,AX
doinit:     MOV AH,9                 ;PRINT ANZ3
            LEA DX,anz3
            INT 21H
            ;
            MOV DX,basisport        ;RTS SETZEN
            ADD DX,4
            MOV AL,3
            OUT DX,AL
            ;
            MOV DX,basisport        ;CTS-SIGNALPORT
            ADD DX,6
            ;
            XOR SI,SI               ;SCHLEIFENZÄHLER
            ;
wartel:     IN AL,DX                 ;"NULLER SCHLUCKEN"
            TEST AL,16
            JZ wartel
            ;
warte2:     INC vergleich_1          ;Länge von "1" abtasten
            IN AL,DX
            TEST AL,16
            JNZ warte2
            ;
            INC SI                   ;12 MAL "1" AUFSUMMIEREN
            CMP SI,15
            JB wartel
            ;
            MOV AX,vergleich_1       ;VERGLEICHSWERT ERRECHNEN
            XOR DX,DX
            ;
            MOV BX,22                ;15/22*DURCHSCHNITTliche LÄnge von "1"
            DIV BX
            MOV vergleich_1,AX
            SHL AX,1
            SHL AX,1
            SHL AX,1
            SHL AX,1
            SHL AX,1
            MOV vergleich_4,AX       ;32 MAL VERGLEICH

```

```

;
del1:  MOV CX,10
del2:  MOV DX,0
      DEC DX
      JNE del2
      DEC CX
      JNE del1
;
geschwtest  RET                ; ZURÜCK
      ENDP
;
lesezeichen PROC NEAR
;
      MOV AX,700H                ; BILDSCHRIMFENSTER LÖSCHEN
      MOV CX,0900H
      MOV DX,184FH
      MOV BH,7
      INT 10H
;
      MOV AH,2                    ; COURSOR POSITIONIEREN
      MOV DX,0900H
      MOV BH,0
      INT 10H
;
      MOV AH,9                    ; PRINT ANZ6
      LEA DX,anz6
      INT 21H
;
lese:     MOV SI,0                ; PUFFER LÖSCHEN
      MOV puffer[SI],0
      MOV DI,0                    ; BITZÄHLER
;
      MOV DX,basisport            ; CTS-REGISTER FERSTELLEN
      ADD DX,6
;
warte3:   IN AL,DX                ; AUF ERSTES BIT WARTEN
      TEST AL,16
      JZ warte3
;
warte4:   CMP DI,25
      JNE h1
      OR BYTE PTR puffer[SI],1
n1:       CMP DI,23
      JNE n2
      OR BYTE PTR puffer[SI],2
n2:       CMP DI,21
      JNE n3
      OR BYTE PTR puffer[SI],4
n3:       CMP DI,19
      JNE n4
      OR BYTE PTR puffer[SI],8
n4:       CMP DI,17
      JNE n5
      OR BYTE PTR puffer[SI],16
n5:       CMP DI,15
      JNE n6
      OR BYTE PTR puffer[SI],32
n6:       CMP DI,13
      JNE n7
      OR BYTE PTR puffer[SI],64
n7:       CMP DI,11
      JNE nichz
      OR BYTE PTR puffer[SI],128
;

```



```

nichz:      IN AL,DX          ; "1"
            TEST AL,16
            JNZ nichz
            INC DI           ;BITZÄHLER ERHÖHEN
            ;
            MOV BX,0         ;NULLZEITZÄHLER
warte5:     INC BX
            CMP BX,vergleich_4
            JAE endezeichen
            IN AL,DX
            TEST AL,16
            JZ warte5
            ;
            SUB BX,vergleich_1 ;ANZAHL NULLBITS FESTSTELLEN
            JB warte4        ;KEIN NULLBIT
            ;
            INC DI           ;BITZÄHLER ERHÖHEN
            SUB BX,vergleich_1
            JB warte6
            ;
warte6:     INC DI           ;BITZÄHLER ERHÖHEN
            CMP DI,27        ;SCHON ALLE BITS GELESEN (27)
            JAE nwar4        ;FALLS NICHT NOCHMAL
            JMP warte4
            ;
nwar4:      INC SI           ;PUFFERZEIGER ERHÖHEN
            JMP lese
            ;
endezeichen: INC SI         ;PUFFERZEIGER ERHÖHEN
            MOV BX,8
eee:        MOV CX,0
ddd:        IN AL,DX        ;KOMMEN NOCH ZEICHEN ???
            TEST AL,16
            JZ nineu
            JMP lese
nineu:      DEC CX
            JA ddd
            DEC BX
            JA eee
            ;
            MOV puffinhalt,SI
            ;
            IN AL,21H        ;TIMER-INTERRUPT ZULASSEN
            AND AL,254
            OUT 21H,AL
            ;
lesezeichen: RET
            ENDP
druck:      PROC NEAR      ;AUSGABE DER ZEICHEN
            ;
            MOV AH,9         ;PRINT ANZ8
            LEA DX,anz8
            INT 21H
            ;
schl_schr:  XOR SI,SI
            CMP SI,puffinhalt
            JAE ende_dru
            MOV AL,puffer[SI]
            INC SI
            ;
            CMP AL,27        ;ESCAPE ???
            JE auswahl
            JMP textausg

```

```

ende_dru:      ;
               MOV puffinhalt,0
               RET

auswahl:      ;
               MOV AL,puffer[SI]           ;HIER BEHANDLUNG DER ESCAPE-CODES
               INC SI
               CMP AL,200
               JBE graphausg               ;GRAPHIK AUSDRUCKEN
               ;
               JMP schl_schr

textausg:     ;
               CMP AL,4
               JE new_line
               CMP AL,10
               JE new_line
               CMP AL,141
               JE pfeil
               CMP AL,139
               JE ungleich
               JMP nanana

new_line:     ;
               XOR DX,DX                   ;NEWLINE MIT NORMALEM VERT-TAB
               MOV AX,10
               INT 17H
               MOV AX,27
               INT 17H
               MOV AX,51
               INT 17H
               MOV AX,33
               INT 17H
               MOV AL,13
               JMP nanana

pfeil:        ;
               MOV AL,45
               XOR DX,DX
               XOR AH,AH
               INT 17H
               MOV AX,8
               XOR DX,DX
               INT 17H
               MOV AL,">"
               JMP nanana

ungleich:     ;
               MOV AL,"="
               XOR DX,DX
               XOR AH,AH
               INT 17H
               MOV AX,8
               XOR DX,DX
               INT 17H
               MOV AL,"/"
               JMP nanana

nanana:       ;
               CMP AL,128
               JB normal_dru
               MOV BL,AL                   ;AB CHR(128) UMSETZTABELLE
               SUB BL,128
               XOR BH,BH
               MOV AL,umsetztab[BX]

normal_dru:   ;
               XOR DX,DX                   ;DRUCKER 1
               MOV AH,0                   ;AUSDRUCKEN
               INT 17H
               ;

```

```

JMP schl_schr
;
graphausg: XOR AH,AH           ; HIER GRAPHIK-AUSGABE
            MOV DI,AX
            ;
            MOV AX,27         ; DRUCKER AUFGRAPHIK VORBEREITEN
            XOR DX,DX
            INT 17H
            MOV AX,75
            INT 17H
            MOV AX,DI
            XOR AH,AH
            INT 17H
            MOV AX,DI
            MOV AL,AH
            XOR AH,AH
            INT 17H
            ;
gr_dr:      MOV BL,puffer[SI]  ; GRAPHIK-DATEN AUSDRUCKEN
            XOR AL,AL
            TEST BL,128
            JZ nna
            OR AL,1
nna:        TEST BL,64
            JZ nnb
            OR AL,2
nnb:        TEST BL,32
            JZ nnc
            OR AL,4
nnc:        TEST BL,16
            JZ nnd
            OR AL,8
nnd:        TEST BL,8
            JZ nne
            OR AL,16
nne:        TEST BL,4
            JZ nnf
            OR AL,32
nnf:        TEST BL,2
            JZ nng
            OR AL,64
nng:        TEST BL,1
            JZ nnh
nnh:        OR AL,128

            XOR AH,AH
            XOR DX,DX
            INT 17H
            INC SI
            DEC DI
            JNE gr_dr
            ;
            MOV AX,27         ; VERT-TAB VERRINGERN
            XOR DX,DX
            INT 17H
            MOV AX,51
            INT 17H
            MOV AX,24
            INT 17H
            ;
            JMP schl_schr
            ;
druck      ENDP
            ;

```



```

code          ENDS
;
daten        SEGMENT PARA
basisport    DW 0
vergleich_1   DW 0
vergleich_4   DW 0
puffinhalt   DW 0
anz1         DB "
               DB " HEWLETT-PACKARD 82240A PRINTER EMULATION
               DB " FOR IBM XT/AT + IBM PROPRINTER II
               DB "
               DB " (C)1989 by Hardy Lau
               DB "
               DB "
               DB " Version 1.2a Tett nang, den 6.Juni.1989
               DB "
               DB " Einen Moment...",10,13,"$"
anz2         DB 10," Bitte wählen Sie aus: 1 ---> COM1",10,13
               DB " " 2 ---> COM2",10,13,"$"
anz3         DB 10,10," Bitte senden Sie jetzt ein Zeichen",10,13
               DB " " um das Programm an die",10,13
               DB " " Rechnergeschwindigkeit anzupassen",10,10,13,"$"
anz4         DB 10,13," Nochmal (J/N) ? $"
anz6         DB 10,13," Rechner Empfangsbereit...",10,13,10,"$"
anz7         DB 10,13,"$"
anz8         DB " " Rechner druckt...",10,10,13,"$"
anz9         DB 10,10,13," Programm beendet...",10,10,13,"$"
umsetztab    DB " ÷*J|Σ πδ≤≥±α μ °",174,175,195,179,253,253,252
               DB " ",",",",95,""
               DB " ",252,252," ",143,142,144,144,144,173,139,"?",",",",^"
               DB " ",247,151,150,156,"-",152,152,248,128,135,165,164,173
               DB 168,15,156,157,21,159,155,131,136,147,150,160,130,162
               DB 163,133,138,149,151,132,137,148,129,143,140,48,146,134
               DB 161,237,145,142,141,153,154,144,139,225,147,143,143,133
               DB 9,"d","f","f",229,229,153,148," ",",",163,156,156,"P"
               DB " ",250,230,227," ",",",172,171," ",167,174,219,175,241," "
puffer       DB 3000 DUP (?)
daten        ENDS
;
stapel       SEGMENT PARA STACK
stapel       DB 2000 DUP (?)
stapel       ENDS
END

```