

INHALT

Datum und Uhrzeit

Sonderzeichen

- beliebige Hexwerte
- extra oder kein Zeilenvorschub

Schalt- und Diagnosefunktionen

- CapsLock, NumLock, ScrollLock
- momentanes Verzeichnis
- Tastaturpuffer löschen
- Kommentar anfügen
- Mausprüfung, Druckertest

Returncode-Erzeugung

- Returncode setzen
- Returncode aus Datum und Zeit
- Returncode aus Tastaturpuffer leer/nicht leer
- Returncode aus Tasteneingabe
- Returncode anzeigen
- Returncode aus CapsLock, NumLock und ScrollLock
- Returncode vom aktuellen Laufwerk

Erzeugung von Tönen (und Wartezeiten)

Interne Textbausteine

Aufruf zum Eingeben einer Zeichenfolge

Einfügen einer Zeichenfolge aus einer Datei

Standardverzeichnis für (temporäre) WECHO-Dateien

Rechnen mit WECHO

Mausmenü

Cursor-Beeinflussung

Beispiele für Batch-Programme mit WECHO

- 255.BAT
- NAME.BAT
- JN.BAT (JA/NEIN)
- GRUSS.BAT
- CAP.BAT
- TESTMENU.BAT

Weitergabe des Programms

Systemvoraussetzungen

(C) Claus-Jürgen Claussen, Riedwiesenweg 10, D-69181 Leimen
claussen.leimen@t-online.de

Probieren Sie die folgenden Aufrufe:

```
WECHO /?      oder WECHO      = Hilfe-Bildschirm
WECHO #       = erzeugt eine Leerzeile
WECHO :Heute ist #W, der #X. #M #Y. Es ist #H.#I Uhr.
WECHO :#W, #X. #M #Y, #H.#I Uhr > Dateiname.xxx
```

Ersatzparameter beginnen stets mit #

```
#Y Jahr      #M Monat      #W Wochentag  #X Tag des Monats
#H Stunde    #N Monat (als Zahl)  #T Sekunde (einstellig)
              #I Minute     #S Sekunde    #Z 1/100 Sekunde
```

In Batch-Dateien auch %-Parameter:

```
WECHO :Text %3 Text #H %NAME% Text #M #T >> Dateiname.xxx
```

WECHO.COM hat eine erheblich erweiterte Funktionalität im Vergleich zum Batch-Befehl ECHO. Das Programm arbeitet mit Ersatzparametern, die stets mit # beginnen. %-Parameter von DOS funktionieren ebenfalls, auch das Zeichen > zum Umleiten von Texten in Dateien. WECHO arbeitet mit vielen Funktionen auch in den DOS-Fenstern von Windows-Betriebssystemen.

Beispiel für eine Eingabe:

```
WECHO Kommentar 1:Heute ist #W, der #X. #M #Y. #;Kommentar 2
```

Hierbei bedeuten:

Kommentar 1: Die Kommandozeile wird vor dem ersten Doppelpunkt nicht ausgewertet. Hier kann also ein Kommentar stehen.

Heute... Auszugebender Text mit Ersatzparametern

#;Kommentar 2 Nach #; wird nicht mehr ausgewertet. Hier kann ein weiterer Kommentar stehen.

#-Ersatzparameter können in beliebiger Anzahl in eine Kommandozeile eingebaut werden. Sie bewirken spezielle Textausgaben oder andere Funktionen.

Datum und Uhrzeit (#\$. erst ab Version 1.33)

```
-----
#W Wochentag, z.B. Dienstag          #H Stunde, z.B. 17 oder 3
#$W Wochentag, z.B. DIE               #$H Stunde, z.B. 17 oder 03
#X Tag des Monats, z.B. 25 oder 7     #I Minute, z.B. 25 oder 05
#$X Tag des Monats, z.B. 25 oder 07   #$I Minute, z.B. 25 oder 5
#M Monat, z.B. September              #S Sekunde, z.B. 13 oder 05
#$M Monat, z.B. SEP                  #T Sekunde, z.B. 10 oder 5
#N Monat, z.B. 12 oder 9
#$N Monat, z.B. 12 oder 09
#Y Jahr, z.B. 1994 oder 2005          #Z 1/100 Sek. z.B. 78 oder 03
#$Y Jahr, z.B. 94 oder 05             #$Z 1/100 Sek. z.B. 78 oder 3

#J Stunde 1...12 (amerikanische Zeit), ab Version 1.1
#$J Stunde 01...12 (amerikanische Zeit), ab Version 1.33
#U gibt a oder p aus für am, a.m., pm, p.m., ab Version 1.1
#$U gibt A oder P aus für AM, A.M., PM, P.M., ab Version 1.33
#G Datum und Uhrzeit komplett in Kurzform
```

Sonderzeichen

```
-----
#00 bis #FF Hexwert für ein auszugebendes Zeichen,
              zum Beispiel für DOS-reservierte Zeichen:

              #3E >   #7C |   #1B <ESC>   #0C FF (Seitenvorschub)
              #3C <   #25 %   #09 <TAB>   #07 BEL (Klingelzeichen)
              #26 &   #34 "   #5E ^       #40 @
              #0D CR allein   #0A LF allein
```

```
#V Zeilenvorschub (CR+LF)
#P am Zeilenende kein Zeilenvorschub
## einmal # ausgeben
```

Schalt- und Diagnosefunktionen

```
-----
#_G schaltet CapsLock auf ON          (bei Win32 ohne Funktion)
#_H schaltet CapsLock auf OFF          (bei Win32 ohne Funktion)
#_N schaltet NumLock auf ON            (bei Win32 ohne Funktion)
#_O schaltet NumLock auf OFF           (bei Win32 ohne Funktion)
#_S schaltet ScrollLock auf ON         (bei Win32 ohne Funktion)
#_T schaltet ScrollLock auf OFF        (bei Win32 ohne Funktion)
#_A schaltet alle 3 Tasten auf OFF     (bei Win32 ohne Funktion)
#_ \ gibt momentanes (aktuelles) Laufwerk aus (ab V 1.32)
#_ \ gibt momentanes (aktuelles) Verzeichnis aus
#- löscht den Tastaturpuffer
#; wertet ab hier die Kommandozeile nicht weiter aus
   (ab hier nur noch Kommentar)

#=M Mausprüfung, mit Kommentar, mit Returncode
   Code 0 Maus ist betriebsbereit
   Code 1 Maus nicht gefunden
   Code 2 Maustreiber nicht gefunden
#=N wie #=M ohne Kommentar, mit Returncode

#=P Druckertest LPT1: mit Kommentar, mit Returncode
   Code 0 Drucker ist betriebsbereit
   Code 1 Drucker ist nicht betriebsbereit oder nicht vorhanden
#=Q wie #=P jedoch LPT2:
#=R wie #=P jedoch LPT3:
#=T Test LPT1: ohne Kommentar, mit Returncode
#=U Test LPT2: ohne Kommentar, mit Returncode
#=V Test LPT3: ohne Kommentar, mit Returncode
```

Returncode-Erzeugung (Errorlevel)

Returncodes können an jeder Stelle der Kommandozeile erzeugt werden.
Werden in einer Zeile mehrmals Returncodes erzeugt, so bleibt der
letzte erhalten und kann mit IF ERRORLEVEL ausgewertet werden.

Returncode setzen: #R00 bis #RFF (Hexwert ergibt den Code 0..255).

Returncode aus dem Wochentag (0...6, 0=Sonntag): #RW
aus dem Tag des Monats (1...31): #RX
aus dem Monat (1...12): #RM
aus dem Jahr (80...199 = 1980...2099): #RY
aus der Stunde (0...23): #RH
aus der Minute (0...59): #RI
aus der Sekunde (0...59): #RT
aus der 1/100 Sekunde (0...99): #RZ

Tastaturpuffer löschen und Returncode erzeugen: #R-
Code 0 = Puffer war leer
Code 1 = Puffer enthielt Zeichen

Returncode aus einem Tastenaufruf erzeugen:

#K erwartet eine Tasteneingabe und erzeugt folgenden Returncode:

CTRL-A bis CTRL-Z 1...26 (CR = 13, TAB = 9, BS = 8, ESC = 27)
Leerzeichen bis Z (ASCII-Tabelle) 32...90 (a...z wie A...Z)

ä,Ä = 142 <F1> bis <F10> = 101 bis 110
ö,Ö = 153 Ins = 182 CursorUp = 172 PgUp = 173 CtrlPgUp = 232
û,Û = 154 Del = 183 CursorDn = 180 PgDn = 181 CtrlPgDn = 218
CtrlCurLeft = 215 CurLeft = 175 Home = 171 CtrlHome = 219
CtrlCurRight = 216 CurRight = 177 End = 179 CtrlEnd = 217

#L wie #K, druckbares Zeichen wird angezeigt

Anmerkung: #K und #L löschen zunächst den Tastaturpuffer, um sicher-
zustellen, daß ein wirklicher Tastenaufruf erfolgt und eine Aus-
wertung einer zuvor betätigten Taste nicht möglich ist.

#RL Returncode in Zahlenform anzeigen (im laufenden Programm)

Beispiel: WECHO :#RFE#RL (FE entspricht 254 dez., es wird also
Returncode 254 erzeugt und als 254 an-
gezeigt)

#RK Returncode in Buchstabenform anzeigen (im laufenden Programm)

#RG wenn Returncode = 1, CapsLock = ON } Abfrage von CapsLock,

#RN wenn Returncode = 1, NumLock = ON } NumLock und

#RS wenn Returncode = 1, ScrollLock = ON } ScrollLock

#R\ Laufwerk über Returncode ermitteln (A = 1, B = 2 usw.)

Erzeugung von Tönen

#[xxx] Für xxx werden bestimmte Zeichen eingesetzt (mindestens eins) [bedeutet Musikanfang,] bedeutet Musikende. Die Zeichen zwischen [und] bedeuten folgendes:

Tondauervorgabe: 1 Takt = 1/18,2 s = ca. 50 ms)

0 = 1 Takt	5 = 8 Takte	{ = 48 Takte	` = 256 Takte
1 = 2 Takte	6 = 12 Takte	} = 64 Takte	
2 = 3 Takte	7 = 16 Takte	~ = 96 Takte	
3 = 4 Takte	8 = 24 Takte	\ = 128 Takte	
4 = 6 Takte	9 = 32 Takte	^ = 192 Takte	(nicht Win32)

Wenn die Tondauervorgabe weggelassen wird, so gilt als Grundeinstellung 5 (8 Takte).

Tonhöhenvorgabe:

	Leerzeichen (space)	oder _ (Unterstrich):	0 Hz (Tonpause)	
	! 77,782 Hz	A 220,00 Hz	S 622,25 Hz	k 1760,0 Hz
(nicht Win32)	" 82,401 Hz	B 233,08 Hz	T 659,26 Hz	l 1864,7 Hz
	\$ 87,307 Hz	C 246,94 Hz	U 698,46 Hz	m 1975,5 Hz
(nicht Win32)	& 92,499 Hz	D 261,63 Hz	V 739,99 Hz	n 2093,0 Hz
	' 97,999 Hz	E 277,18 Hz	W 783,99 Hz	o 2217,5 Hz
	(103,826 Hz	F 293,66 Hz	X 830,61 Hz	p 2349,3 Hz
) 110,000 Hz	G 311,13 Hz	Y 880,00 Hz	q 2489,0 Hz
	* 116,541 Hz	H 329,63 Hz	Z 932,30 Hz	r 2637,0 Hz
	+ 123,471 Hz	I 349,23 Hz	a 987,80 Hz	s 2793,8 Hz
	, 130,813 Hz	J 369,99 Hz	b 1046,5 Hz	t 2960,0 Hz
	- 138,591 Hz	K 392,00 Hz	c 1108,7 Hz	u 3136,0 Hz
	. 146,832 Hz	L 415,30 Hz	d 1174,7 Hz	v 3322,4 Hz
	/ 155,563 Hz	M 440,00 Hz	e 1244,5 Hz	w 3520,0 Hz
	: 164,814 Hz	N 466,16 Hz	f 1318,5 Hz	x 3729,3 Hz
	; 174,614 Hz	O 493,88 Hz	g 1396,9 Hz	y 3951,1 Hz
	= 184,997 Hz	P 523,25 Hz	h 1480,0 Hz	z 4186,0 Hz
	? 195,998 Hz	Q 554,37 Hz	i 1568,0 Hz	
	@ 207,652 Hz	R 587,33 Hz	j 1661,2 Hz	

Beispiel für ein paar Töne mit Pause: WECHO :#[4ACE5BD2_3F]
Beispiel für eine Wartezeit (ca. 1 s) ohne Ton: :WECHO #[8_]
Bei den Zeichen zwischen [und] wird zwischen Groß- und Kleinbuchstaben unterschieden, wie die vorhergehende Tabelle zeigt.

Laufende Tonfolgen bzw. Wartezeiten können durch das Bedienen einer beliebigen Taste abgebrochen werden. Bei Abbruch wird ein Returncode erzeugt (gleicher Code wie bei #K).
Anmerkung: #[xxx] löscht zunächst den Tastaturpuffer, um sicherzustellen, daß durch eine bereits zuvor betätigte Taste kein ungewollter Abbruch erfolgt.

Interne Textbausteine (vgl. auch Beispiel NAME.BAT)

#Qx internen Textbaustein einfügen,
für x können alle Buchstaben von a bis z stehen
#QA @ECHO OFF CR/LF SET NAME=
#QB @ECHO OFF CR/LF
#QC IF ERRORLEVEL
#QD IF NOT ERRORLEVEL
#QE bis #QZ noch nicht festgelegt

Aufruf zum Eingeben einer Zeichenfolge (string)

#:x Für x können die Ziffern 0...9 und die Buchstaben a bis z stehen. Die Funktion erwartet die Eingabe einer Zeichenfolge (max. 128 Zeichen) über die Tastatur, die mit <Enter> abgeschlossen wird. Die Zeichenfolge wird dann in der folgenden Datei abgespeichert:

C:\WECHO\WECHO_x.BAT (x=0...9, a...z)

Das Unterverzeichnis C:\WECHO muß vorhanden sein. Die letzten Zeichen in der Datei sind CR/LF. Wenn diese Datei als Batch-Datei lauffähig ist, kann sie z.B. aus einem gerade laufenden Batch-Programm mit CALL aufgerufen werden.

Einfügen einer Zeichenfolge aus einer Datei

#*x Für x können die Ziffern 0...9 und die Buchstaben a bis z stehen. Der Inhalt der Datei

C:\WECHO\WECHO_x.BAT (x=0...9, a...z)

wird eingefügt. Eventuelle CTRL-Z werden nicht mit eingefügt. Es wird die gesamte Datei eingefügt (keine Begrenzung). TABs werden in eine entsprechende Anzahl Leerzeichen gewandelt. Um Probleme zu vermeiden, sollten TABs und Ctrl-Z gar nicht erst benutzt werden.

Beim Einfügen werden Kleinbuchstaben in Großbuchstaben gewandelt. #*x ab Version V 1.2

#.x Für x können die Ziffern 0...9 und die Buchstaben a bis z stehen. Der Inhalt der Datei

C:\WECHO\WECHO_x.BAT (x=0...9, a...z)

wird eingefügt. Eventuelle CTRL-Z werden nicht mit eingefügt. Es wird die gesamte Datei eingefügt (keine Begrenzung). TABs werden in eine entsprechende Anzahl Leerzeichen gewandelt. Um Probleme zu vermeiden, sollten TABs und Ctrl-Z gar nicht erst benutzt werden. Kleinbuchstaben werden nicht in Großbuchstaben gewandelt. #.x ab Version V 1.2

#+x Für x können die Ziffern 0...9 und die Buchstaben a bis z stehen. Der Inhalt der Datei

C:\WECHO\WECHO_x.BAT (x=0...9, a...z)

wird als Textstring eingefügt. Eventuelle CR, LF, CTRL-Z werden nicht mit eingefügt. Es werden nur max. die ersten 128 Zeichen der Datei eingefügt. TABs werden in eine entsprechende Anzahl Leerzeichen gewandelt. Um Probleme zu vermeiden, sollten TABs und Ctrl-Z gar nicht erst benutzt werden. Keine Wandlung von Klein- in Großbuchstaben. Da #+x alle CR und LF entfernt, eignet es sich zum Abschneiden dieser Steuerzeichen von Textstrings. Zu beachten ist, daß WECHO stets einmal CR/LF am Ende abgibt, wenn es nicht ausdrücklich mit #p unterbunden wird.

#~x Für x können Buchstaben A bis Z stehen. Für die Funktionen #:x, #*x, #.x und #+x sind normalerweise feste Dateinamen in einem festen Verzeichnis auf dem Laufwerk C: vorgegeben, z.B. C:\WECHO\WECHO_5.BAT. Das Laufwerk C: ist also vordefiniert.

Mit dem Parameter #~x (x = A bis Z) kann die Laufwerksdefinition für die genannten Dateien auf ein anderes Laufwerk umbenannt werden. Die Umbenennung gilt solange, bis sie durch eine erneute Umbenennung überschrieben wird oder bis eine neue WECHO-Zeile gestartet wird.

Eine neue WECHO-Zeile verwendet bis zu einer Umbenennung Laufwerk C. Die Umbenennung macht es möglich, z.B. ein Verzeichnis E:\WECHO\.. für temporäre Dateien zu verwenden, wobei E: vorteilhafterweise ein RAM-Laufwerk sein kann. Dies ergibt Geschwindigkeitsvorteile und entlastet die Festplatte C: z.B. bei Schleifen in einer Batch-Datei von sich wiederholenden Zugriffen. Vergessen Sie nicht, auf RAM-Laufwerken, die Sie mit WECHO verwenden wollen, das Verzeichnis \WECHO\ einzurichten. Das muß bei jedem Booten erneut geschehen. Natürlich kann auch #~A verwendet werden, um WECHO-Dateien auf dem Diskettenlaufwerk A: (Verzeichnis A:\WECHO\..) zu benutzen.

#~x ab Version V 1.2

Ab WECHO Version 1.3 gibt es die Möglichkeit, das Laufwerk für WECHO-Dateien über die DOS-Systemumgebung vorzudefinieren. Die Eintragung erfolgt mit dem SET-Befehl von DOS. Soll z.B. das RAM-Laufwerk E: für die WECHO-Dateien zuständig sein, so werden am besten direkt ins AUTOEXEC.BAT die beiden folgenden Zeilen hinzugefügt:

```
SET WECHODRV=E
MD E:\WECHO
```

WECHO überprüft die Eintragung bei WECHODRV und übernimmt die Laufwerksangabe. Wird in der Kommandozeile mit #~x umdefiniert, so gilt diese Definition. Beim nächsten Aufruf von WECHO gilt dann wieder die Angabe unter WECHODRV.

Ein Tip: Die DOS-Systemumgebung kann leicht zu klein werden, wenn mit SET mehrere Eintragungen vorgenommen werden. Die Systemumgebung läßt sich mit folgender (gekennzeichneter) Eintragung in CONFIG.SYS auf 1024 Byte vergrößern:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS
DOS=HIGH,UMB
LASTDRIVE=K
---> SHELL=C:\COMMAND.COM /P /E:1024
COUNTRY=049,437,C:\DOS\COUNTRY.SYS
DEVICEHIGH=C:\DOS\ANSI.SYS
BREAK=OFF
FILES=50
BUFFERS=40,8
STACKS=18,512
DEVICEHIGH=C:\DOS\RAMDRIVE.SYS 44
DEVICE=C:\DOS\RAMDRIVE.SYS 256 512 512 /E
```

```
WECHO :...
#{ Vorzeichen_1  Operand_1  Operator  Vorzeichen_2  Operand_2  }
WECHO :...
#{      -           45           /           -           32      }
```

Regeln:

- Stand: 4. März 2005

#o0...#o9 Mausmenu (siehe auch Beispiel TESTMENU.BAT) ab V 1.3

Bei Aufruf von #o0...#o9 (MAUSMENU) wird ein Mauscursor auf den Bildschirm gesetzt. Durch Anklicken eines Zeichens auf dem Bildschirm wird dessen ASCII-Code als Returncode erzeugt und das Unterprogramm beendet.

Steuer-, Leer- und Graphikzeichen (ASCII-Code größer als 165) werden beim Anklicken ignoriert. Der Parameter nach dem "o" muß eine Ziffer (0...9) sein. Diese Ziffer bestimmt den Bereich, in dem sich der Mauscursor bewegen läßt. Beispiel: WECHO :#o4

0 = Bewegungsfreiheit auf dem gesamten Bildschirm

1...9 = Anzahl der Zeilen, auf denen sich der Cursor bewegen läßt.

Gleichzeitig wird die horizontale Bewegung auf 10 Zeichen begrenzt.

Das Menu, in dem sich die Maus bewegen soll, muß vorher mit ECHO oder TYPE auf den Bildschirm geschrieben werden. Testen Sie die Mausbewegung mit der Batch-Datei TESTMENU.BAT. In dieser Batch-Datei wird das Batch-Programm ERRLEVEL zur Anzeige des Returncodes aufgerufen.

Wenn Sie auf die ESC-Taste drücken, während der Mauscursor auf dem Bildschirm ist, wird das Menu beendet und der Returncode 255 erzeugt.

Returncode 255 wird ebenfalls erzeugt, wenn

- die Maus oder der Maustreiber fehlt
- nach dem "o" von #o... keine Ziffer folgt.

#(...) Cursor-Beeinflussung ab V 1.33

Bei Aufruf von #(...) (Cursor-Beeinflussung) wird der Cursor in seinem Aussehen und/oder in seiner Position auf dem Bildschirm beeinflusst. Zwischen den Klammern kann stehen:

#(ON) oder #(N) Cursor wird eingeschaltet (VGA und höher)
#(OFF) oder #(F) Cursor wird ausgeschaltet (VGA und höher)
#(VGA) oder #(V) Standardeinstellung VGA und höher (auch MCGA)
#(CGA) oder #(C) Standardeinstellung CGA und höher
#(MDA) oder #(M) Standardeinstellung MDA, HERCULES, EGA
#(BIG) oder #(B) Großer Cursor
#(-) Dünner Cursor
#(+) Dicker Cursor
#(X46) Setzt Cursor in die 46. Spalte (geht von 1 bis 80)
#(Y17) Setzt Cursor in die 17. Zeile (geht von 1 bis 25)

Es können mehrere Parameter in einem Kommando angegeben werden, z.B.
WECHO :#(OFF X53 Y12)Hallo#(ON BIG)

Beispiele für Batch-Programme mit WECHO:

```
@echo off
rem 255.BAT erzeugt die Datei 255.BIN, die alle Zeichen von
rem 000 bis 255 (00 bis FFH) enthält
WECHO :#00#01#02#03#04#05#06#07#08#09#0a#0b#0c#0d#0e#0f#p>255.bin
WECHO :#10#11#12#13#14#15#16#17#18#19#1a#1b#1c#1d#1e#1f#p>>255.bin
WECHO :#20#21#22#23#24#25#26#27#28#29#2a#2b#2c#2d#2e#2f#p>>255.bin
WECHO :#30#31#32#33#34#35#36#37#38#39#3a#3b#3c#3d#3e#3f#p>>255.bin
WECHO :#40#41#42#43#44#45#46#47#48#49#4a#4b#4c#4d#4e#4f#p>>255.bin
WECHO :#50#51#52#53#54#55#56#57#58#59#5a#5b#5c#5d#5e#5f#p>>255.bin
WECHO :#60#61#62#63#64#65#66#67#68#69#6a#6b#6c#6d#6e#6f#p>>255.bin
WECHO :#70#71#72#73#74#75#76#77#78#79#7a#7b#7c#7d#7e#7f#p>>255.bin
WECHO :#80#81#82#83#84#85#86#87#88#89#8a#8b#8c#8d#8e#8f#p>>255.bin
WECHO :#90#91#92#93#94#95#96#97#98#99#9a#9b#9c#9d#9e#9f#p>>255.bin
WECHO :#a0#a1#a2#a3#a4#a5#a6#a7#a8#a9#aa#ab#ac#ad#ae#af#p>>255.bin
WECHO :#b0#b1#b2#b3#b4#b5#b6#b7#b8#b9#ba#bb#bc#bd#be#bf#p>>255.bin
WECHO :#c0#c1#c2#c3#c4#c5#c6#c7#c8#c9#ca#cb#cc#cd#ce#cf#p>>255.bin
WECHO :#d0#d1#d2#d3#d4#d5#d6#d7#d8#d9#da#db#dc#dd#de#df#p>>255.bin
WECHO :#e0#e1#e2#e3#e4#e5#e6#e7#e8#e9#ea#eb#ec#ed#ee#ef#p>>255.bin
WECHO :#f0#f1#f2#f3#f4#f5#f6#f7#f8#f9#fa#fb#fc#fd#fe#ff#p>>255.bin
```

```
@echo off
rem NAME.BAT Erfragt Ihren Namen und trägt ihn unter
rem          NAME=.... in die Systemumgebung ein.
rem          Nachschauen mit DOS-Befehl SET
if not exist c:\WECHO\nul md c:\WECHO
if exist C:\WECHO\xhelp.bat del C:\WECHO\xhelp.bat
if exist C:\WECHO\WECHO_0.bat del C:\WECHO\WECHO_0.bat
WECHO :Bitte geben Sie Ihren Namen ein: #:0
WECHO :#QA#+0>C:\WECHO\xhelp.bat
call C:\WECHO\xhelp
if exist C:\WECHO\xhelp.bat del C:\WECHO\xhelp.bat
if exist C:\WECHO\WECHO_0.bat del C:\WECHO\WECHO_0.bat
echo Name wurde eingetragen
:ende
```

```

@echo off
rem JN.BAT Tastenaufwurf ausprobieren
WECHO :Bitte drücken Sie eine Taste: #L#v
goto start
:oben
WECHO :#K#p
:start
if errorlevel 13 if not errorlevel 14 goto creturn
if errorlevel 27 if not errorlevel 28 goto escape
if errorlevel 74 if not errorlevel 75 goto ja
if errorlevel 78 if not errorlevel 79 goto nein
WECHO :#[4ACM]Nur mit J, N, ESC oder ENTER kommen Sie zum Ziel#p#0d
goto oben
:creturn
WECHO :#vMit ENTER ans Ziel
goto ende
:escape
WECHO :#vMit ESC hat's geklappt
goto ende
:ja
WECHO :#vJ wie JA ist eine Lösung
goto ende
:nein
WECHO :#vMit N wie Nein gelingt der Ausstieg
goto ende
:ende

@echo off
rem GRUSS.BAT Ein Spruch zu jeder Tageszeit
WECHO :#rh
if errorlevel 22 goto nacht
if errorlevel 18 goto abend
if errorlevel 12 goto tag
if errorlevel 10 goto vormit
if errorlevel 8 goto morgen
goto frueh
:nacht
WECHO :Es ist #h.#i Uhr. So spät nachts sollte man nicht am Computer sitzen.
goto ende
:abend
WECHO :Abends um #h.#i Uhr ist die beste Arbeitszeit.
goto ende
:tag
WECHO :Um #h.#i Uhr, am hellen Tag am Computer?
goto ende
:morgen
WECHO :Es ist #h.#i Uhr. Morgens schon so munter?
goto ende
:vormit
WECHO :Es ist #h.#i Uhr. Vormittags schon so arbeitsam?
goto ende
:frueh
WECHO :Es ist #h.#i Uhr. So früh kann noch kein Werk gelingen.
goto ende
:ende
WECHO Nur eine Extra-Leerzeile

```

```

@echo off
rem CAP.BAT Diese Datei holt %1 bis %9 von der Kommandozeile und
rem schreibt sie unter den Namen CP1 bis CP9 in die Systemumgebung.
rem Hier können die Parameter (in Großbuchstaben gewandelt) als
rem %CP1% bis %CP9% benutzt werden, z. B. für IF "CP1"=="A".....
WECHO :#qbSET CP1=%1#vSET CP2=%2#vSET CP3=%3#vSET CP4=%4#vSET
CP5=%5>c:\WECHO\WECHO_A.bat
WECHO :SET CP6=%6#vSET CP7=%7#vSET CP8=%8#vSET CP9=%9>>c:\WECHO\WECHO_A.bat
WECHO :#*A#p>help.bat
call help
del help.bat

```

```

@echo off
REM TESTMENU.BAT (Mausmenu auf dem Bildschirm)
rem ANSI.SYS muß in CONFIG.SYS installiert sein. Sonst funktionieren
rem die Farben und der Bildaufbau nicht. Die Programme WECHO.COM
rem und ERRLEVEL.BAT müssen im Pfad liegen oder im aktuellen
rem Verzeichnis.

```

```

echo
WECHO :
echo
WECHO :
WECHO :
WECHO :
WECHO :
WECHO :
WECHO :
echo
WECHO :#o6
call errlevel
echo
WECHO :
echo
echo
echo
echo
WECHO :#o2
call errlevel

```

#1b[46;30m BITTE MIT DER MAUS AUSWÄHLEN #1b[0m		
#1b[41;37mJJJ#1b[0m	JA	
#1b[42;30mNNN#1b[0m	NEIN	
#1b[43;37mQQQ#1b[0m	QUIT	
#1b[44;37mEEE#1b[0m	EXIT	
#1b[45;37mAAA#1b[0m	Laufwerk A:	
#1b[46;30mBBB#1b[0m	Laufwerk B:	

JA / NEIN	J N Q E	QUIT / EXIT
LW A: B:	A B C D	Laufwerk C: D:

Weitergabe des Programms

Das Programm WECHO.COM ist Freeware. Es darf zu nichtgewerblichen Zwecken von jedermann kostenlos genutzt werden. Irgendwelche Haftung seitens des Autors besteht nicht. Die Weitergabe ist erlaubt und erwünscht, jedoch nur zusammen mit dieser Beschreibung. Eine gewerbliche Nutzung bedarf der Zustimmung des Autors.

Bei Auslieferung waren noch die Programme TE.COM und TA.COM mit dabei. Sie dienen dazu, herauszufinden, welche Tastencodes bei der Bedienung der Tastatur erzeugt werden.

Die ebenfalls mitgelieferte Batch-Datei ERRLEVEL.BAT kann Ihnen nützlich sein. Sie bringt nach Ablauf eines Programms den hinterlassenen Returncode auf den Bildschirm. ERL.BAT macht eigentlich das gleiche, bringt jedoch den gefundenen Returncode unter dem Namen ERL in die Systemumgebung.

TA.COM, TE.COM und ERRLEVEL.BAT sind Public-Domain-Software. Jede Art von Nutzung ist gestattet.

Vielleicht macht Ihnen auch die Benutzung des Programms RECHNE Spaß. Ist Freeware. Die Rechenfunktionen davon sind in WECHO mit eingebaut, unter WECHO :#{.....}.

Systemvoraussetzungen

Sie brauchen einen IBM-kompatiblen XT/AT-Rechner im Textmodus (hat jeder). Prozessor ab 8088/86 aufwärts. DOS ab 3.30.

Es ist notwendig, ANSI.SYS zu laden. Der normale Lautsprecher des Computers genügt für die Ausgabe von Tönen.

Schreiben Sie mir, wenn Ihnen das Programm zusagt, wenn es Sie geärgert hat oder wenn Sie Anregungen haben. Die Adresse finden Sie zu Beginn dieser Beschreibung.

Viel Spaß beim Ausprobieren!

