ATARI®

ATARI PROGRAMM
RXG 7075

ATARI MICROSOFT BASIC II

REFERENZ-TABELLE

Program Copyright © 1982 by
Microsoft - All rights reserved

Für die deutsche Übersetzung

© 1983 Jegliche Rechte vorbehalten
ATARI ELEKTRONIK - Vertriebsges. mbH

ATARINSIDE

ATARI PROGRAMM
RXG 7075

ATARI MICROSOFT BASIC II

REFERENZ-TABELLE

Program Copyright © 1982 by
Microsoft - All rights reserved

Für die deutsche Übersetzung

© 1983 Jegliche Rechte vorbehalten
ATARI ELEKTRONIK - Vertriebsges. mbH

ATARINSIDE

INHALT

BEFEHLE	3
ANWEISUNGEN	5
NUMERISCHE FUNKTIONEN	10
ALPHANUMERISCHE (STRING) FUNKTIONEN	11
SPEZIELLE FUNKTIONEN	12
BESONDERHEITEN DES ATARI PRIVAT COMPUTER SYSTEMS	13
FEHLERMITTEILUNGEN	15

ATARINSIDE

PROGRAMM BEFEHLE

BEFEHL	BEISPIELE	BEDEUTUNG
AUTO	AUTO AUTO 10,10	die Numerierung des Programmes erfolgt automatisch. Ohne Zusatz beginnt diese Numerierung bei 100 und erhöht sich in 10er Schritten. Im 2. Beispiel beginnt die Numerierung bei 10 und erhöht sich in 10er Schritten.
CLOAD	CLOAD	lädt ein Programm im tokenized Format von Kassette in den Computer-Speicher.
CONT	CONT	setzt die Programmausführung nach STOP oder BREAK fort.
CSAVE	CSAVE	speichert das im Speicher befindliche Programm im tokenized Format auf eine Kassette.
DEL	DEL 10 DEL 10-100 DEL 10- DEL -100	löscht eine oder einen bestimmten Bereich an Programmzeilen. Verwenden Sie Bindestriche, um den Bereich zu bestimmen.
DOS	DOS	ruft Disketten-Operations-System-Menü auf. Um wieder zum Microsoft BASIC II zu gelangen, geben Sie B ein.
KILL	KILL "D:TEST.BAS"	löscht das angegebene Programm von der Diskette. Im Beispiel das Programm mit dem Namen TEST.BAS.
LIST	LIST LIST 10 LIST 10-100 LIST 10- LIST -100 LIST "P:" LIST "C:"	listet eine, einen Bereich oder alle Programmzeilen auf den Bildschirm. Mit Bindestrichen können Sie den Bereich bestimmen. Mit LIST können Programmteile auch auf Drucker, Kassettenrecorder oder einem anderen Peripheriegerät ausgegeben werden.
LOAD	LOAD "D:BEISPIEL.1" LOAD	lädt ein Programm im tokenized Format in den Speicher des Computers. Es kann auf Diskette oder Kassette sein.
LOCK	LOCK "D:TEST.1"	schützt das angegebene Programm auf Diskette vor versehentlichem Löschen.
MERGE	MERGE "D:TEST.1" MERGE "C:"	mischt das angegebene Programm mit dem bereits im Speicher stehenden. Dabei auftretende gleiche Zeilennummern werden überschrieben.

PROGRAMM BEFEHLE

BEFEHL	BEISPIELE	BEDEUTUNG
NAME.TO	NAME"D:ALT"TO"NEU"	benennt das Programm mit dem Namen "ALT" in das Programm "NEU" um.
NEW	NEW	löscht das im Speicher befindliche Programm.
RENUM	RENUM RENUM 100,50,10	numeriert bestehende Programmzeilen neu. Ohne Zusatz beginnt die Neunummerierung bei Zeile 10 und wird um jeweils 10 erhöht.Das zweite Beispiel gibt der Zeile 50 die Nummer 100 und erhöht dann jeweils um 10.
RUN	RUN RUN 100	startet die Programmausführung bei der niedrigsten oder der angegebenen Zeilennummer.
SAVE	SAVE"D:BEISPIEL.1"	speichert das Programm unter dem angegebenen Namen im tokenized Format auf einem Peripheriegerät.
SAVE,, LOCK	SAVE"D:TEST"LOCK	durch die Zusatzanweisung LOCK kann das Programm nach dem Einladen nicht mehr gelistet oder verändert werden.
TROFF	TROFF	schaltet die TRACE-Funktion aus.
TRON	TRON	schaltet die TRACE-Funktion ein. Mit Trace wird die jeweils abgearbeitete Zeilennummer angezeigt.
UNLOCK	UNLOCK"D:TEST.1"	hebt die Blockierung eines Programmes wieder auf, damit es überschrieben, umbenannt oder gelöscht werden kann.
VERIFY	VERIFY"D:TEST.1" VERIFY"C:"	vergleicht das ausgewählte Programm mit dem im Speicher befindlichen Programm. Wird eine Abweichung entdeckt, erfolgt die Meldung "TYPE MISMATCH ERROR".

ATARINSIDE

PROGRAMM ANWEISUNGEN

ANWEISUNG	BEISPIELE	BEDEUTUNG
AFTER	AFTER(3600)10	zählt die in Klammern angegebene Zeit in 1/50 Sekunden und verzweigt zur angegebenen Zeilennummer (hier 10).
CLEAR	CLEAR	löscht alle Variablen und setzt alle Stringlängen auf Null.
CLEAR STACK	CLEAR STACK	setzt alle Zeitzähler auf Null zurück.
CLOSE	CLOSE #1	schließt den angegebenen Datenkanal.
COMMON	COMMON ALL COMMON A,X$	ermöglicht die übernahme von Variablen von einem Programm in ein anderes. Der Zusatz bestimmt dabei welche Variablen übernommen werden.
DEF	DEF A(X,Y)=X/Y	erlaubt das Definieren eigenener Funktionen. Im Beispiel wird A(X,Y) als X/Y festgelegt. Zum Definieren eigener STRING-Funktionen wird die Erweiterung benötigt.
DIM	DIM A$(120) DIM A(10,20)	reserviert einen Bereich, in den dann die Werte der Variablen geschrieben werden. Auch mehrdimensionale Felder können reserviert werden.
END	END	beendet das Programm, schließt offene Files und setzt alle Zeitzähler auf 0.
ERROR	ERROR 2	simuliert den angegebenen Fehler. Wird benutzt um das Verhalten des Programms im Fehlerfall zu testen.
FOR..TO.. STEP / NEXT	FOR I=1 TO 10	wiederholt diesen Programmteil solange, bis die Variable, (hier I), den angegebenen Wert erreicht hat. Der Zusatz STEP gibt an, um wieviel die Variable nach jeder Ausführung erhöht wird.
GET	GET #1,X GET #1,AT (8,2);x	liest einzelne Bytes und weist sie einer Variablen zu (hier X).
GOSUB / RETURN	GOSUB 100	verzweigt das Programm zur angegebenen Programmzeile und führt nach dem RETURN die den GOSUB folgende Anweisung aus.
GOTO	GOTO 1020	verzweigt das Programm zur angegebenen Zeilennummer.

PROGRAMM ANWEISUNGEN

ANWEISUNG	BEISPIELE	BEDEUTUNG
IF...THEN	IF A=30 THEN A=30-X	vergleicht Variablen oder Konstanten miteinander oder mit einem Wert. Trifft der Vergleich zu, erfolgt die Ausführung der hinter dem Vergleich stehenden Anweisung. Sonst wird das Programm in der nächsten Zeile fortsetzt.
IF...THEN ...ELSE	IF A=30 THEN A=30-X ELSE A=A+1	entspricht der Anweisung IF...THEN. Das Programm führt jedoch bei nicht zutreffendem Vergleich die hinter ELSE stehende Anweisung aus.
INPUT	INPUT I INPUT I$ INPUT"Zahl";I INPUT"NAME";N$	nimmt Werte und Text von Tastatur und Diskette / Kassette auf und weist sie einer Variablen zu.
INPUT...AT	INPUT AT(2,3) I$	erfüllt die gleichen Funktionen wie INPUT mit Angabe der Zeile und Reihe auf dem Bildschirm.
LET	LET Z=4.5	weist einer Variablen einen Wert zu.
LINE INPUT	LINE INPUT "NAME:";N$	ist wie INPUT, erlaubt jedoch auch die Eingabe von Interpunktionen. Die Angabe eines Strings (i.e. "NAME:") ermöglichtes, anstatt des "?" den 'Prompt' selber zu bestimmen.
LINE INPUT ...AT	LINE INPUT AT(4,4)I$	erfüllt die gleiche Funktion wie die Anweisung INPUT...AT mit LINE INPUT.
MOVE	MOVE 55,222,5	bewegt Teile des Speicherbereichs von einer Adresse zur anderen. Im Beispiel werden 5 Bytes von der Adresse 55 zur Adresse 222 bewegt. Die Angabe der Werte kann auch in hex erfolgen.
NEXT	NEXT NEXT I NEXT I,J,X	steht am Ende eines FOR...TO...NEXT Blocks. Die Angabe der Variablen ist nicht unbedingt erforderlich.
NOTE	NOTE#4,I,J	ermittelt den Sektor und das Byte, die als nächstes von der Diskette gelesen werden, und weist diese Werte den zwei Variablen zu.

ATARINSIDE

PROGRAMM ANWEISUNGEN

ANWEISUNG	BEISPIELE	BEDEUTUNG
ON ERROR	ON ERROR 500	verzweigt Programm im Fehlerfall zur angegebenen Zeilennummer. Durch die Anweisung RESUME wird die Ausführung des Programms fortgesetzt.
ON...GOSUB / RETURN	ON A GOSUB 100,200,300	führt ein Unterprogramm aus, je nachdem welchen Wert der angegebene Ausdruck ... hat. In dem Beispiel wird bei einem Wert der Variablen A von 1, 2 oder 3 die Zeile 100, 200 oder 300 angesprungen.
ON...GOTO	ON A GOTO 100,200,300	ist identisch mit der Anweisung ON... GOSUB, nur erfolgt die Verzweigung mit GOTO.
OPEN	OPEN #3,"P:" OUTPUT OPEN #5,"D:TT" INPUT	öffnet einen Datenkanal zum Lesen oder Schreiben von Daten. Der Anweisung muß die Art der Datenübertragung folgen: INPUT = Eingabe OUTPUT = Ausgabe UPDATE = Ein- und Ausgabe APPEND = Anhängen von Daten
OPTION BASE	OPTION BASE1	legt fest, ob die Indizierung der Variablenfelder mit 1 oder 0 beginnt. Erfolgt keine OPTION BASE-Anweisung, wird 0 als Wert angenommen.
OPTION CHR	OPTION CHR1 OPTION CHR2 OPTION CHR0	reserviert Speicherplatz für einen eigenen Zeichensatz. OPTION CHR1 reserviert 1024 Byte. OPTION CHR2 reserviert 512 Byte. OPTION CHR0 löscht die Reservierung.
OPTION PLM	OPTION PLM1 OPTION PLM2 OPTION PLM0	reserviert Speicherplatz für Player-Missile-Grafik. Die Anzahl der Bytes entspricht jeweils denen der OPTION CHR-Anweisung.
OPTION RESERVE	OPTION RESERVE 24	reserviert Speicherplatz angegebener Menge für eigene Anwendungen, wie Maschinensprache-Unterroutinen.
PRINT	PRINT "Hallo" PRINT 25*4 PRINT R$? Y	gibt Texte oder den Wert von Variablen auf dem Bildschirm aus. Es kann auch ein Fragezeichen statt PRINT eingegeben werden.
PRINT..AT	PRINT AT(1,5)I	wie PRINT. Nur erfolgt die Ausgabe an einer definierten Stelle des Bildschirms.

PROGRAMM ANWEISUNGEN

ANWEISUNG	BEISPIELE	BEDEUTUNG
PRINT..SPC	PRINT SPC(5)"Boot"	gibt angegebene Anzahl von Leerstellen vor Text oder Variablen aus. Gezählt wird ab der Cursorposition.
PRINT..TAB	PRINT TAB(5)"HUND"	beginnt mit der Ausgabe von Text und Variablen an der angegebenen Position. Gezählt wird immer vom linken Rand.
PRINT USING	? USING "##";1 ? USING "##.##" ? USING "###,###.##" ? USING "xx###.##" ? USING "DM###.##" ? USING "$####.##" ? USING "xx$###.##" ? USING "##^^^^" ? USING "+###" ? USING "###-" ? USING "!" ? USING "% %"	gibt Zahlen rechtsbündig aus. fügt Dezimalpunkt ein. setzt alle 3 Stellen ein Komma. füllt führende Leerstellen mit x. setzt vor die Zahl DM. setzt gleitendes Dollarzeichen($). setzt gleitendes $ und x. gibt Zahl mit Exponent aus. setzt Vorzeichen vor Zahl. setzt Minuszeichen nach Zahl, falls Zahl negativ. gibt erstes Zeichen vom STRING aus. gibt Teile eines STRINGS aus.
PUT	PUT #6,123	gibt einzelne Bytes an den geöffneten Datenkanal aus.
RANDOMIZE	RANDOMIZE RANDOMIZE 30	gibt dem Zufallszahlengenerator eine andere Anfangszahl an, ab der Zufallszahlen erzeugt werden.
READ / DATA	READ A,B$,C DATA 12,"an",3	liest Daten aus einer DATA-Anweisung und weist sie der Variablen zu.
REM	REM Kommentar ! Kommentar ' Kommentar X=1:REM mit : X=1!ohne :	erlaubt es Ihnen, Ihre Kommentare zu dem erstellten Programm einzugeben, um es übersichtlicher zu machen.

ATARINSIDE

PROGRAMM ANWEISUNGEN

ANWEISUNG	BEISPIELE	BEDEUTUNG
RESTORE	RESTORE RESTORE 100	gestattet die Wiederbenutzung von DATA-Anweisungen; daß Einlesen dieser Zeilen beginnt wieder von vorne. Ist die Zeilennummer angegeben, wird bei der nächsten READ-Anweisung mit dieser Zeile begonnen.
RESUME	RESUME RESUME 55 RESUME NEXT	verzweigt das Programm nach einer Fehlerroutine wieder in den Programmfluß. Ohne Zusatz ist es die Zeile, in der der Fehler auftrat. Mit Zusatz einer Zahl wird die Zeile festgelegt und mit NEXT folgt die nächste Anweisung.
RETURN	RETURN	führt aus einem Unterprogramm und setzt das Programm mit der Anweisung nach GOSUB fort.
STACK	PRINT STACK IF STACK=0 THEN X=20	gibt an, wie oft der Zeitzähler für die Anweisung SOUND oder AFTER noch benutzt werden kann (max. 20mal).
STOP	STOP	stoppt die Ausführung des Programms. Durch die Anweisung CONT wird es mit der nächsten Zeile fortgesetzt.
WAIT...AND	WAIT &D40B, &FF,110	wartet mit der Programmausführung bis die angegebenen Adressen, im Beispiel D40B undiert mit FF, einen bestimmten Wert erreicht haben (110).

PROGRAMM FUNKTIONEN (NUMERISCHE)

FUNKTION	BEISPIELE	BEDEUTUNG
ABS	Y=ABS(-7)	errechnet den Betrag.
ATN	X=ATN(5.3)	errechnet den Arcustangens der Zahl.
COS	PRINT COS(.95)	errechnet den Kosinus.
EXP	EULER=EXP(3)	errechnet die Exponentialfunktion (e in x-te Potenz).
INT	? INT(3.6)	gibt die größte ganze Zahl aus, die kleiner ist als die angegebene (hier 3).
LOG	L=LOG(5)	errechnet den natürlichen Logarithmus einer positiven Zahl.
RND	R=RND PRINT RND(0) NR=RND(100)	erzeugt eine Zufallszahl mit einfacher Genauigkeit. Ist hier keine Ziffer in Klammern angegeben, liegt die erzeugte Zahl zwischen 0 und 1. Wird eine Zahl angegeben, liegt der Wert zwischen 0 und der Zahl (hier 0 - 100) und ist ganzzahlig.
SGN	R=SGN(R*B)	gibt das Vorzeichen einer Zahl aus. Ist sie positiv hat SGN den Wert von +1. Wenn sie negativ ist, den Wert -1. Ist sie Null, hat auch SGN den Wert 0.
SIN	PRINT SIN(1)	errechnet den Sinus.
SQR	W=SQR(25)	errechnet die Quadratwurzel einer positiven Zahl.
TAN	A=TAN(.22)	errechnet den Tangens einer Zahl.
+	A$=B$+C$	verbindet zwei Stringvariablen.
ASC	PRINT ASC("SAM")	gibt die ATASCII-Codezahl des ersten Zeichens des in Klammern gesetzten Textes aus. Im Beispiel 83 ("S").
CHR$	A$=CHR$(65)	wandelt die in Klammern stehende Zahl des ATASCII-Code in das entsprechende Zeichen um und weist es der String-Variablen zu.
INKEY$	R$=INKEY$	gibt die zuletzt gedrückte Taste an.

ATARINSIDE

PROGRAMM FUNKTIONEN (ALPHANUMERISCHE)

FUNKTION	BEISPIELE	BEDEUTUNG
INSTR	A=INSTR(X,A$,B$)	sucht einen kleineren String in einem größeren ab einer bestimmten Stelle. Im Beispiel wird B$ im A$ gesucht. Ist er enthalten, wird der Variablen A die Position zugewiesen, andernfalls 0. Fehlt die Angabe der Stelle, ab der gesucht werden soll, so beginnt die Suche an der ersten Stelle. Beispiel: A$="MAUS":B$="AUS" A hat den Wert 2, weil AUS ab der 2. Stelle in MAUS enthalten ist.
LEFT$	PRINT LEFT$("LAUS",3)	gibt die linken Stellen des Strings an. Im Beispiel wird LAU ausgegeben.
LEN	? LEN(A$)	gibt die Länge des Strings an (Anzahl der Zeichen).
MID$	PRINT MID$(A$,4,6)	gibt Zeichen aus der Mitte des Strings aus. Die erste Zahl bestimmt dabei ab welcher Stelle das geschieht und die zweite gibt die Anzahl der Zeichen an.
RIGHT$	PRINT RIGHT$(A$,6)	gibt die rechten Zeichen des String aus. Die Zahl gibt dabei an, wieviel Zeichen ausgegeben werden. Im Beispiel die letzten sechs Zeichen.
SCRN$	C$=SCRN$(1,6) PRINT ASC(C$)	liest den ASCII-Code eines Zeichens vom Bildschirm. Ist ein Grafikmodus aufgerufen, wird der Farbwert gelesen.
STR$	X$=STR$(9.99)	weist den Wert einer Variablen einem String zu.
STRING$	? STRING$ (36,"x") ? STRING$ (10,65)	gibt ein bestimmtes Zeichen mit einer festgelegten Anzahl von Wiederholungen aus. Im ersten Beispiel werden 36 Sterne ausgegeben, im zweiten Beispiel 10 mal der Buchstabe A.
TIME$	PRINT TIME$	gibt die Zeit im Format Std Std:Min Min:Sek Sek aus.
VAL	PR=VAL(A$)	weist die Ziffern eines Strings einer Variablen zu.

PROGRAMM FUNKTIONEN (SPEZIELLE)

FUNKTION	BEISPIELE	BEDEUTUNG
EOF	EOF(4)	gibt an, ob das zuletzt angesprochene IOCB eine EOF-Marke erreicht hat.
ERL	PRINT ERL	gibt die Zeilennummer an, in der der letzte Fehler aufgetreten ist.
ERR	PRINT ERR	gibt die Codenummer des letzten Fehlers an.
FRE(0)	PRINT FRE(0)	gibt den noch zur Verfügung stehenden Speicherplatz in Bytes an.
PEEK	? PEEK(1536) AA=PEEK(CHR)	liest den Inhalt der in den Klammern angegebenen Adresse im Speicherbereich. Die Angabe der Adresse kann als Hex- oder Dezimalzahl erfolgen. Die Adresse kann im RAM oder ROM liegen.
POKE	POKE 1034,34 POKE AA,CHR POKE A,Y+2	weist einer Adresse im Speicher einen bestimmten Wert zu (0 - 255).
STATUS	? STATUS(4)	liest das vierte Byte eines IOCB's. Dieses teilt einen Fehler mit (1=Fehler, 0=kein Fehler bei Ein- oder Ausgabe).
USR	X=USR(560,4)	wird zum Aufrufen von Maschinensprache-Unterprogrammen benutzt. Im Beispiel ist diese ab der Adresse 560 im Speicher abgelegt. Die zusätzliche Zahl wird von diesem Unterprogramm benutzt.
VARPTR	A=VARPTR(A$) ? VARPTR(A$)+1 P=VARPTR(PLM1) A=VARPTR(CHR1) X=VARPTR (RESERVE)	gibt die Speicheradresse an, ab der ein reservierter Speicherplatz beginnt. Dieser wird mit OPTION xxx oder DIM reserviert. Der Zusatz in den Klammern entspricht alternativ denen der OPTION Anweisungen.

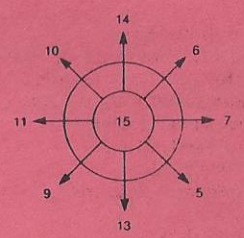
ATARINSIDE

BESONDERHEITEN DES ATARI PRIVAT COMPUTER SYSTEMS

GRAFIK	BEISPIELE	BEDEUTUNG
CLS	CLS CLS 25	löscht den Textbildschirm und wechselt die Farbe des Hintergrunds in den mit der Zahl angegebenen Wert. Bei einem Grafikbildschirm ohne Textfenster wird die Farbe des Randes gewechselt.
COLOR	COLOR 4	gibt das Farbregister und somit die Farbe an, in der Grafiken erstellt werden.
FILL	FILL 5,5 TO 10,10	füllt einen bestimmten Bereich des des Bildschirms mit einer Farbe aus.
GRAPHICS	GRAPHICS 0 GRAPHICS 2+16 GRAPHICS 7+32	wählt einen der Grafikmodi aus. Durch den Zusatz +16 wird kein Textfenster freigehalten. +32 verhindert das Löschen des Bildschirms.
PLOT PLOT...TO	PLOT X,Y PLOT X,Y TO X1,Y1	zeichnet eine Linie oder einen Punkt an eine vorbestimmte Stelle auf den Bildschirm.
SETCOLOR	SETCOLOR 5,4,3	bestimmt die Farbe und Helligkeit des angegebenen Farbregister. Im Beispiel bedeutet: 5 die Nummer des Farbregisters (0-8). 4 die gewählte Farbe (0-15). 3 die Helligkeit der Farbe (0-14). Je höher die Zahl, desto heller ist die Farbe.
DREHREGLER /PADDLE CONTROLLERS	DREHREGLER(0)=624 DREHREGLER(1)=625 DREHREGLER(2)=626 DREHREGLER(3)=627 DREHREGLER(4)=628 DREHREGLER(5)=629 DREHREGLER(6)=630 DREHREGLER(7)=631	Die jeweils aktuellen Werte der Drehregler stehen in den Speicheradressen 624 bis 631. Jeder der Eingänge des ATARI-Computers nimmt je zwei Drehregler auf. Der Wert des linken Drehreglers steht in der Adresse 624, der des zweiten in der Adresse 625 usw. Die Werte liegen zwischen 1 und 228 und werden mit PEEK (xxx) gelesen.

BESONDERHEITEN DES ATARI PRIVAT COMPUTER SYSTEMS

SPIEL-STEUERUNG	ADRESSE	BEDEUTUNG
KNÖPFE DER DREHREGLER	PTRIG(0)=636 PTRIG(1)=637 PTRIG(2)=638 PTRIG(3)=639 PTRIG(4)=640 PTRIG(5)=641 PTRIG(6)=642 PTRIG(7)=643	Die Werte der Knöpfe der Drehregler stehen in den Adressen von 636 bis 643. Ist ein Knopf gedrückt, enthält die jeweilige Adresse den Wert 0, sonst den Wert 1. Gelesen werden die Werte mit PEEK.
STEUER-KNÜPPEL	STICK(0)=632 STICK(1)=633 STICK(2)=634 STICK(3)=635	Die Werte der Steuerknüppel stehen in den Adressen 632 bis 635. Welche Werte jeweils den Bewegungen entsprechen, entnehmen Sie bitte der Abbildung.
FEUERKNÖPFE DER STEUER-KNÜPPEL	STRIG(0)=644 STRIG(1)=645 STRIG(2)=646 STRIG(3)=647	Der Wert in der jeweiligen Adresse ist 0 wenn der Feuerknopf gedrückt ist, sonst 1.
KONSOLEN-Tasten	OPTION -> PEEK(53279)=3 SELECT -> PEEK(53279)=5 START -> PEEK(53279)=6 Keine -> PEEK(53279)=7	In der Adresse 53279 steht der Wert der jeweils gedrückten Konsolentaste. Dieser Wert ist solange unverändert, wie die jeweilige Taste gedrückt ist.



ATARINSIDE

FEHLERMITTEILUNGEN

Fehler Nummer	Bedeutung
1	NEXT ohne FOR.
2	Syntaxfehler
3	RETURN ohne GOSUB.
4	keine weiteren Daten vorhanden.
5	unmögliche Funktion aufgerufen.
6	Ergebnis zu groß / zu klein.
7	Speicher voll.
8	nicht vorhandene Zeilennummer.
9	nicht groß genug DIMensioniert.
10	zweimal gleiche Variable DIMensioniert.
11	es wurde versucht, durch Null zu teilen.
12	Anweisung darf nicht direkt eingegeben werden.
13	Buchstaben bei numerischer Variable eingegeben.
14	genereller Ein- / Ausgabefehler.
15	eingegebener String zu lang (max. 255 Zeichen)
16	Funktion zu komplex, muß aufgeteilt werden.
17	Fortsetzen der Programmausführung unmöglich.
18	falsche USR-Funktion eingegeben.
19	keine RESUME Anweisung nach Fehlerfall.
20	RESUME vor ON ERROR eingegeben.
21	FOR ohne NEXT.
22	Programm ist gegen Listen geschützt.
23	unzulässiger Befehl nach AFTER-Anweisung.
128	BREAK-Taste wurde gedrückt
129	der gewählte Datenkanal ist bereits geöffnet
130	das gewählte Gerät existiert nicht
131	Datenkanal ist nur zum Schreiben geöffnet
132	unvollständige / falsche Anweisung
133	Datenkanal / Datei nicht geöffnet
134	falsche Datenkanal-Nummer
135	Datenkanal ist nur zum Lesen geöffnet
136	Versuch über das Dateiende zu lesen
137	Datei abgeschnitten
138	Gerät antwortet nach einiger Zeit nicht
139	Negatives / nicht erfolgtes Quittungssignal vom Gerät
140	Lesefehler auf seriellem Bus
141	Cursor befindet sich außerhalb des Bildschirms
142	Computer verarbeitet eintreffende Daten zu langsam
143	Prüfsummen Fehler
144	Gerätefehler
145	Unzulässiger Bildschirm-Modus
146	dem Datenkanal wurde eine unbekannte Anweisung gegeben
147	für den gewählten Grafikmodus zu wenig Speicherplatz
160	nicht vorhandenen Diskettenstation angesprochen
161	zu viele offenen Diskettendateien
162	Diskette ist vollgeschrieben
163	grober Lese-/Schreibfehler
164	Dateinummern / Sektoren vertauscht
165	falscher Dateiname
166	unzulässiger POINT-Befehl
167	Datei blockiert (LOCK-Befehl)
168	unzulässiger Diskettenbefehl
169	Direktory dieser Diskette ist voll (64 Namen)
170	gewählte Datei befindet sich nicht auf der Diskette
171	unzulässiger POINT-Befehl

ATARINSIDE

ATARI®
A Warner Communications Company

Wichtige Informationen

Lieber Computerfreund, lieber Kunde, lieber Händler!

Jeder, der sich einmal selbst damit beschäftigt hat, ein Computerprogramm zu fertigen, weiß, welche Arbeit und geistige Mühe aufgewendet werden muß, um eine Problemlösung zu finden und sie anwenderfreundlich zu programmieren. Die Erfüllung dieser Voraussetzungen erfordert viel Erfahrung und hohe finanzielle und zeitliche Investitionen. Das Ergebnis sind gute und erfolgreiche Computerprogramme, die von interessierten Anwendern nachgefragt werden und deshalb für den Händler verkäuflich sind.

Diese Tatsache machen sich einige dadurch zunutze, daß sie die mit hohen Voraufwendungen geschaffenen erfolgreichen Programme der Firma Atari kopieren oder ihren Kunden die Möglichkeit anbieten, die gewünschten Programme auf Diskette zu überspielen. Sie meinen, damit ihren Kunden ein gutes und billiges Angebot zu machen. Die Kunden wissen jedoch meist nicht, daß sie lediglich ein vermeintlich gutes und billiges Angebot erhalten.

Abgesehen davon, daß das Angebot zur Überspielung von Programmen und das Anbieten und Verkaufen illegal kopierter Programme strafrechtlich verboten ist, weil es sich dabei um Verletzungen des Urheberrechtes (COMPUTERPROGRAMM PIRATERIE) handelt, die von Atari gegenüber jedermann ohne Ansehen der Person gerichtlich verfolgt wird, so ist auch die Annahme falsch, das Angebot sei günstig oder billig:

- Gestohlene Ware ist immer billig. Der Dieb hat keine Voraufwendungen. Er eignet sich nur fremdes Eigentum an, für die der Käufer keine Gewährleistung erhält.
- Der Händler, der das Kopieren von Programmen anbietet, anstatt Originale zu verkaufen schmarotzt an fremder Leistung.
- Der interessierte Kunde wird bald keine guten Programme mehr kaufen können und illegale Programme wird der Handel bald auch nicht mehr anbieten können.

Letzteres deswegen, weil niemand mehr bereit und in der Lage sein wird, gute verkaufsfähige Programme zu entwickeln, wenn nicht die Möglichkeit besteht, die hohen Voraufwendungen durch Verkäufe wieder zu verdienen. Die Piraten sind geistig weder in der Lage noch überhaupt bereit, sich der Mühe zu unterziehen, Programme zu entwickeln. Sie können und wollen nur durch Diebstahl fremder guter Leistung eine schnelle bequeme Mark verdienen.

Wer also Interesse daran hat, daß das Angebot an guten Computerprogrammen wächst, sollte die illegalen „billigen" Angebote meiden und mit dazu beitragen, daß den Totengräbern der Computer-Programmentwicklung und damit des Computerhandels das Handwerk gelegt wird.

Wir danken für Ihr Verständnis und freuen uns über jeden Hinweis von Ihnen.

Atari Elektronikvertriebsges. mbH

ATARI ELEKTRONIK Vertriebsgesellschaft mbH · Bebelallee 10 · 2000 Hamburg 60 · Tel. 040/511 80 91

ATARINSIDE

ATARI®

A Warner Communications Company

ATARI-Elektronik Vertriebsgesellschaft mbH
Postfach 60 01 69 · Bebelallee 10 · 2000 Hamburg 60

Jegliche Rechte vorbehalten.
Vermietung, Verleih, Vervielfältigung
und öffentliche Aufführung verboten.

ATARINSIDE