

ATARI®

ATARI PROGRAMM
RXG 8032
Steckmodul

**REFERENZKARTEN
ATARI LOGO**

(c) 1983 all rights reserved ATARI INC.
Für die deutsche Übersetzung:
(c) 1984 jegliche Rechte vorbehalten
ATARI ELEKTRONIK Vertriebsges. mbH

ATARI PROGRAMM
RXG 8032
Steckmodul

**REFERENZKARTEN
ATARI LOGO**

(c) 1983 all rights reserved ATARI INC.
Für die deutsche Übersetzung:
(c) 1984 jegliche Rechte vorbehalten
ATARI ELEKTRONIK Vertriebsges. mbH

ATARINSIDE

INHALTSVERZEICHNIS

EINFUEHRUNG.....	1
BESCHREIBUNG DER EINGABEN.....	2
BESCHREIBUNG DER OPERANDEN.....	2
ATARI LOGO PRIMITIVES.....	4
1. TURTLE-GRAFIK.....	4
2. WORTE UND LISTEN.....	7
3. VARIABLEN.....	9
4. ARITHMETISCHE OPERATIONEN.....	9
5. DEFINIEREN UND EDITIEREN VON PROZEDUREN.....	10
6. PROGRAMMABLAUFSTEUERUNG UND VERGLEICHEN.....	11
7. LOGISCHE OPERATIONEN.....	12
8. DIE VERBINDUNG ZUR AUSSENWELT.....	12
9. DIE VERWALTUNG DES ARBEITS- SPEICHERS.....	14
10. DATEIEN.....	15
11. SPEZIELLE PRIMITIVKOMMANDOS.....	16
12. SPEZIELLE TASTEN.....	16
INDEX.....	19

ATARINSIDE

EINFUEHRUNG

ATARI LOGO WIRD GESTARTET

1.
Schalten Sie den Computer aus.
2.
Schalten Sie Ihren Monitor oder Fernseher ein.
3.
Falls Sie eine Diskettenstation haben, schalten Sie diese ein und warten, bis die Kontrolllampe nicht mehr leuchtet.
Fahren Sie bei Schritt 5 fort, falls Sie keine Diskettenstation besitzen.
4.
Legen Sie die ATARI MASTER DISK in das Laufwerk der Diskettenstation und schliessen Sie den Schacht der Diskettenstation. Sie koennen auch eine beliebige andere Diskette benutzen, sofern auf ihr ein DOS geschrieben ist.
5.
Setzen Sie das LOGO-Sleckmodul in den Modulschacht Ihres Computers und schalten Sie den Computer ein.

Das Fragezeichen ist das Promtsymbol. Sobald das "?" auf dem Bildschirm erscheint, koennen Sie Ihre Befehle eingeben. Das weisse Rechteck ist der CURSOR. Er zeigt an, an welcher Stelle das naechste eingegebene Zeichen erscheint.

ATARI INSIDE

BESCHREIBUNG DER EINGÄBEN

In diesem Abschnitt werden alle Befehlsbestandteile der LOGO-PRIMITIVES erklärt.

BESCHREIBUNG DER OPERANDEN

byte.....	Eine ganze Zahl im Bereich 0 bis 255.
char, character.....	Buchstabe, Ziffer oder Satzzeichen.
colornumber.....	Eine Zahl im Bereich 0 bis 127.
condnumber.....	Eine ganze Zahl im Bereich 0 bis 21.
degrees.....	Gradzahl eines Winkels. Eine reale Zahl zwischen -9999.9999 und 9999.9999. Dieser Bereich kann mit dem Kommando REPEAT erweitert werden.
device.....	Bezeichnung eines externen Gerätes. "C;" ist der Cassettenrecorder, "D;" Diskettenstation, "P;" der Drucker. Das Anführungszeichen ist stets erforderlich.
distance.....	Eine reale Zahl zwischen -9999.9999 und 9999.9999. Dieser Bereich kann durch das Kommando REPEAT erweitert werden.
duration.....	Eine ganze Zahl im Bereich 0 bis 255.
filename.....	Ein regelrecht gebildeter Dateiname.
freq.....	Eine ganze Zahl im Bereich 14 bis 64.000. Sie gibt die in Hertz gemessene Frequenzhöhe an.
inputs.....	Worte, die von einem Doppelpunkt angeführt werden. Sie werden im Zusammenhang mit TO verwendet.

instruction-list.....

Eine durch LOGO ausführbare Liste von Prozeduren.

joystick-number.....

Eine ganze Zahl im Bereich von 0 bis 3.

list.....

Eine durch eckige Klammern eingeschlossene Befehlsliste.

n,a,b,c,z.....

Eine beliebige Zahl.

name.....

Ein Wort, das eine Prozedur oder eine Variable bezeichnet.

namelist.....

Eine Namensliste.

object, obj.....

Ein LOGO-Bestandteil (Wort, Liste oder Zahl).

paddlenumber.....

Eine ganze Zahl zwischen 0 und 7.

pennumber.....

Eine ganze Zahl zwischen 0 und 2.

position, pos.....

Eine Liste zweier Zahlen, die die Position der Turtle oder des CURSORS beschreibt.

pred.....

Ein logischer Ausdruck, dessen Auswertung entweder TRUE oder FALSE ergibt.

shapenumber.....

Eine ganze Zahl zwischen 0 und 15.

shapespec.....

Eine Liste aus 16 Zahlen, die das Shaperaster darstellt.

turtlenumber.....

Eine ganze Zahl zwischen 0 und 3.

voice.....

0 oder 1.

volume.....

Eine ganze Zahl zwischen 0 und 15.

word.....

Eine Buchstabenfolge, die keinen Zwischenraum enthalten darf.

ATARI SIDE

a + b.....	Ergibt a+b.
a - b.....	Ergibt a-b.
a * b.....	Ergibt a*b.
a / b.....	Ergibt a/b.
a < b.....	Ergibt TRUE, falls a kleiner als b ist.
a > b.....	Ergibt TRUE, falls a grösser als b ist.
a = b.....	Ergibt TRUE, falls a gleich b ist.

5. DEFINIEREN UND EDITIEREN VON PROZEDUREN

EDIT,ED name(s).....	Startet den Editor mit den genannten Prozeduren.
EDNS.....	Startet den Editor mit allen gepeicherten Variablen.
END.....	Beendet eine durch TO eingeleitete Definition.
TO name (inputs).....	Beginnt eine Prozedurdefinition.

6. ABLAUFSTEUERUNGEN UND BEDINGUNGEN

COND condnumber.....	Ergibt TRUE, falls die durch condnumber bezeichnete Bedingung eingetreten ist.
IF pred list1 (list2).....	Falls pred TRUE ergibt, wird list1 ausgefuehrt, sonst list2.
OUTPUT,OP obj.....	Uebergibt die Ablaufsteuerung und den Wert obj an die aufrufende Routine.
OVER turtlenumber....., pennumber.....	Ergibt einen Code, der die Kollision zwischen turtlenumber und pennumber symbolisiert.
REPEAT n list.....	Fuehrt list n-mal aus.
RUN list.....	Fuehrt list aus. Ergebnis ist was von list ausgegeben wird.
STOP.....	Unterbricht die laufende Prozedur und uebergibt die Ablaufkontrolle an das aufrufende Programm.
TOUCHING turtlenumber1..., turtlenumber2.....	Ergibt einen Code, der die Kollision zwischen turtlenumber1 u. turtlenumber2 symbolisiert.
WAIT n.....	Unterbricht fuer n-mal sechzigstel Sekunden den Programmablauf.
WHEN condnumber list.....	Es wird ein WHEN-Demon erzeugt, der fuer die Ausfuehrung von list sorgt, sobald die durch condnumber bezeichnete Bedingung zutrifft.
WHEN condnumber [].....	Hebt die Zuordnung des WHEN-Demons an die Befehlsfolge list auf.

ATARI INSIDE

7. LOGISCHE OPERATIONEN

#AND pred1 pred2.....	Ergibt TRUE, falls alle EingabevARIABLEn TRUE sind.
FALSE.....	Ergibt das Wort FALSE, das als konstante EingabevARIABLE f <u>uer</u> logische OperATIONen verwendet werden kann.
NOT pred.....	Ergibt FALSE, wenn pred TRUE ist und umgekehrt.
#OR pred1 pred2.....	Ergibt TRUE, sobald eine der EingabevARIABLEn TRUE ist.
TRUE.....	Ergibt das Wort TRUE, das als konstante EingabevARIABLE f <u>uer</u> logische OperATIONen verwendet werden kann.

8. BILD VERARBEITUNG ZUR AUSZEICHNUNG

CT.....	Löscht den Textabschnitt des Bildschirms.
FS.....	Reserviert den ganzen Bildschirm f <u>uer</u> Grafik.
JOY joysticknumber.....	Ergibt den momentanen Wert des durch joysticknumber bezeichneten Joysticks.
JOYB joysticknumber.....	Ergibt TRUE, falls der Trigger des durch joysticknumber bezeichneten niedergedrückt ist.
KEYF.....	Ergibt TRUE, falls eine Taste gedrückt wurde, aber noch nicht eingelesen wurde.
PADDLE paddlenumber.....	Ergibt den Drehwinkel des durch paddlenumber bezeichneten Paddles.
PADDLEB paddlenumber.....	Ergibt TRUE, falls der Trigger des durch paddlenumber bezeichneten Paddles niedgedrückt ist.
#PRINT,PR obj.....	Druckt obj gefolgt von einem Zeilenverschub aus (Listen werden ohne eckige Klammern ausgedruckt).

RD.....	Ergibt das vom aktuellen Peripheriegeraet (normalerweise der Tastatur) gelesene Zeichen. Falls kein Zeichen ansteht, wird das Programm bis zur Eingabe eines Zeichens angehalten.
RL.....	Wie RD, liest aber eine ganze Zeile.
SETCURSOR position.....	Setzt den CURSOR auf die angegebene Position.
SETENV voice duration....	F <u>uer</u> den durch voice bezeichneten Tongenerator wird eine Huetkurve definiert, so dass ein durch TOOT erzeugter Ton pro Zeiteinheit, die durch duration bestimmt wird, um eine Lautstaerkestupe verringert wird.
SHOW obj.....	Druckt obj in eckigen Klammern aus.
SS oder CTRL S.....	Toilt den Bildschirm in Grafik- und Textbereich auf.
TOOT voice freq.....	Erzeugt einen Ton der Frequenz freq, der Lautstaerke volume und der Dauer duration ueber den Tongenerator voice.
TS oder CTRL T.....	Stellt den ganzen Schirm f <u>uer</u> Textausgabe zur Verf <u>ugung</u> .
#TYPE obj.....	Druckt obj aus, ohne den CURSOR an den Anfang einer der folgenden Zeilen zu setzen.

ATARINSIDE

9. VERWALTUNG DES ARBEITSSPEICHERS

ERALL.....	Loescht den gesamten Arbeitsspeicher.
ERASE,ER name(s).....	Loescht die genannten Prozeduren.
ERN name(s).....	Loescht die genannten Variablen.
ERNS.....	Loescht alle Variablen.
ERPS.....	Loescht alle Prozeduren.
NODES.....	Ergibt die Anzahl der freien Nodes (Node = Speichereinheit).
PO name(s).....	Druckt die Definition der benannten Prozeduren.
POALL.....	Druckt die Definition der benannten Prozeduren und Variablen.
POD condnumber.....	Druckt die momentan aktiven WHEN-Demons, die die Bedingung condnumber ueberwachen.
PODS.....	Druckt alle aktiven WHEN-Demons.
PONS.....	Druckt die Namen und Werte aller Variablen.
POPS.....	Druckt die Definitionen aller Prozeduren.
POTS.....	Druckt die Namen aller definierten Prozeduren.
RECYCLE.....	Erzwingt eine Garbage Collection.

10. DATEIEN

CATALOG device:.....	CATALOG D: listet die Diskettendirectory auf. CATALOG C: listet alle Prozedurdefinitionen und Namen der Cassettedatei auf.
ERF device:filename,.....	Loescht auf dem Peripheriegeraet device die Datei name.
LOAD device:filename,.....	Laedt vom Peripheriegeraet device die Datei name.
SAVE device:filename,.....	Speichert den Inhalt des Arbeitsspeichers unter dem Namen name auf dem Peripheriegeraet device. Wenn es sich bei device um einen Drucker handelt, werden alle Prozeduren ausgedruckt.
SETREAD device:filename,.....	Vereinbart die Datei filename auf dem Peripheriegeraet device als Eingabedatei fuer die Befehle RC und RL.
SETREAD [],.....	Hebt eine mit SETREAD vereinbarte Dateizuordnung auf.
SETWRITE device:filename,.....	Schreibt alle auf dem Bildschirm ausgedruckten Zeichen in die Datei filename auf dem Geraet device.
SETWRITE [],.....	Hebt die durch SETWRITE vereinbarte Dateizuordnung auf.

11. SPEZIELLE PRIMITIVES

.CALL n.....	Ruft eine ab Adresse n gespeicherte Assemblerroutine auf.
.DEPOSIT n byte.....	Speichert das Datum byte in der Adresse n (dezimal).
.EXAMINE n.....	Ruft den Inhalt der Adresse n (dezimal) auf.
.PRIMITIVES.....	Druckt eine Liste aller LOGO Grundkommandos
.SETSCR n.....	Setzt das Streckungsverhältnis des Bildschirms auf den Wert n.

12. BESONDERE TASTENFUNKTIONEN

ATARI-Taste oder.....	Nach Druecken dieser Taste erscheinen alle Zeichen in Inversdarstellung. Nachmaliges Druecken der Taste hebt die Inversdarstellung auf.
*BREAK.....	Unterbricht alle laufenden Taetigkeiten. Im Editiermodus vorgenommene Aenderungen werden ignoriert, im Direktmodus wird die zuletzt eingegebene Zeile geloescht.
*CTRL→.....	Bewegt den CURSOR eine Stelle nach rechts.
*CTRL←.....	Bewegt den CURSOR eine Stelle nach links.
CTRL↑.....	Bewegt den CURSOR eine Zeile nach oben.
CTRL ↓.....	Bewegt den CURSOR eine Zeile nach unten.

*CTRL 1.....	unterbricht die Bildschirmausgabe bis CTRL 1 erneut eingegeben wird.
*CTRL A.....	Bewegt den CURSOR zum Anfang der Zeile.
*CTRL CLEAR.....	Loescht die Zeile von der CURSOR-Position an bis zum Zeilenende.
*CTRL DELETE BACK SPACE.....	Loescht das Zeichen an der CURSOR-Position.
*CTRL E.....	Bewegt den CURSOR zum Ende der Zeile.
CTRL F.....	Setzt den gesamten Bildschirm auf Grafikmodus.
CTRL INSERT.....	Erzeugt eine Leerzeile. Der restliche Bildschirm wird nach oben geschoben.
CTRL S.....	Teilt den Bildschirm in Grafik- und Textbereich.
CTRL T.....	Setzt den gesamten Bildschirm auf Textmodus.
CTRL V.....	Fuehrt im Editiermodus zu einem Vorschub auf die naechste Bildschirmseite.
CTRL W.....	Zeigt die vorhergehende Bildschirmseite.
CTRL X.....	Bewegt den CURSOR an den Anfang der editierten Prozedur.
*CTRL Y.....	Ruft wahrend des Editierens den Inhalt des Loeschbuffers auf; außerhalb des Editors wird das zuletzt eingegebene Kommando ausgedruckt.
CTRL Z.....	Bewegt den CURSOR an das Ende der editierten Prozedur.

ATARINSIDE

*DELETE BACK SPACE.....	Löscht das Zeichen links vom CURSOR.
ESC.....	Beendet den Editiermodus und kehrt in den Direktmodus zurück.
F1,F2,F3,F4.....	Programmierbare CURSOR-Tasten.
*RETURN.....	Beendet die eingegebene Zeile und setzt den CURSOR an den Anfang der nächsten Zeile.
SHIFT DELETE BACK SPACE.....	Löscht die Zeile von der CURSOR-Position an bis zum Ende.
SHIFT INSERT.....	Erzeugt an der CURSOR-Position eine Leerzeile.
\(backslash).....	Dies Zeichen wird nicht als Kommando sondern als druckbares Zeichen interpretiert, z.B. +, -, .
SYSTEM RESET.....	Startet LOGO neu, der Inhalt des Arbeitsspeichers geht verloren.

I	IF.....11 inputs.....2 instructionlist.....3 INT.....9 INVERS VIDEO key.....16
J	JOY.....12 JOYE.....12 joysticknumber.....3
K	KEYP.....12
L	LAST.....8 LEFT.....5 list.....3 LT.....5 LIST.....8 LISTP.....8 LOAD.....15 LPUT.....8
M	MAKE.....9 MEMBERP.....9
N	name..... namelist..... NAMEP.....9 NODES.....14 NOT.....12 NUMBERP.....9
O	object.....3 OP.....12 OUTPUT, OF.....11 OVER.....11
P	PADDLE.....12 PADDLEB.....12 paddlenumber.....3 PC.....5 PE.....5 PEN.....5 PENDOWN, PD.....5 penumber.....3 PENUPI, PU.....5 PN.....5 PO.....14 POALL.....14 POD.....14 PODS.....14 PONS.....14 POPS.....14 pos.....3 POS..... position.....3 POTS.....14 pred.....3 .PRIMITIVES.....16 PRINT, PR.....12 PRODUCT.....9 PUTSH.....5 PX.....5
R	RANDOM.....9 RD.....17 RECYCLE.....14 REMAINDER.....9 REPEAT.....11 RANDOM.....9 RETURN.....18 RIGHT, RT.....5 RL.....17 ROUND.....9 RUN.....11
S	SAVE "D:".....15 SAVE "P:".....15 SE.....8 SETBG.....6 SETC.....6 SETCURSOR.....13 SETENV.....13 SETH.....6

INDEX

A

AND.....12
ASCII.....7
ASK.....4
ATARI key.....16

B

BACK,BK.....4
backslash.....18
BG.....4
BREAK.....16
BUTFIRST,BF.....7
BUTLAST,BL.....7
byte.....2

C

.CALL.....16
CATALOG.....15
char.....2
CHAR.....7
character.....2
CLEAN.....4
COLOR.....4
colornumber.....2
COND.....11
condnumber.....2
COS.....9
COUNT.....7
CS.....4

CT.....12
CTRL I.....17
CTRL →.....16
CTRL ←.....16
CTRL ↑.....16
CTRL ↓.....16
CTRL CLEAR.....17
CTRL DELETE BACK S.....17

CTRL INSERT.....17
CTRL A.....17
CTRL E.....17
CTRL F.....17
CTRL S.....17
CTRL T.....17

CTRL V.....17
CTRL W.....17
CTRL X.....17
CTRL Y.....17
CTRL Z.....17

D

degrees.....2
DELETE BACK S.....18
.DEPOSIT.....16
device.....2
distance.....2
duration.....2

E

EACH.....4
EDIT,ED.....10
EDNS.....10
EDSH.....4
EMPTYP.....7
END.....10
EQUALP.....7
ERALL.....14
ERASE,ER.....14
ERF.....15
ERN.....14
ERNS.....14
ERFS.....14
ESC.....18
.EXAMINE.....16

F

F1,F2,F3,F4.....18
FALSE.....12
filename.....2
FIRST.....8
FORWARD,FD.....4
FFPUT.....8
frequency, freq.....2
FS(CTRL F).....12

G

GETSH.....5

H

HEADING.....5
HOME.....5
HT.....5

S

SETPC.....	6	WHO.....	7
SETPN.....	6	WINDOW.....	7
SETPOS.....	6	word.....	3
SETREAD.....	15	WORD.....	8
SETREAD [I].....	15	WORDP.....	8
SETSCR.....	16	WRAP.....	7
SETSH.....	6		
SETSP.....	6	X	
SETWRITE.....	15	XOR.....	7
SETWRITE [I].....	15		
SETX.....	6	Y	
SETY.....	6	YCOP.....	7
SHAPE.....	6		
shapenumber.....	3	a + b.....	10
shapespec.....	3	a - b.....	10
SHIFT DELETE BACK S	18	a * b.....	10
SHIFT INSERT.....	18	a / b.....	10
SHOW.....	12	a ^ b.....	10
SHOWNP.....	6	a = b.....	10
SIN.....	9	obj = obj2.....	8
SPEED.....	6		
SORT.....	9		
SS (CTRL S).....	13		
ST.....	6		
STOP.....	11		
SUM.....	9		
SYSTEM RESET.....	18		

T

TELL.....	6
THING.....	9
TO.....	10
TOOT.....	13
TOUCHING.....	11
TRUE.....	12
TS (CTRL T).....	13
turtlenumber.....	3
TYPE.....	13

V

VOICE.....	3
VOLUME.....	3

W

WAIT.....	11
WHEN.....	11
WHEN IT.....	11

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 illegaler 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®



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