

Referenzkarte OMIKRON.Debugger

Die Tastaturbelegung des Debuggers

SHIFT ↑	: Scrollt den Bildschirm (der Cursor bleibt stehen)
SHIFT + CONTROL ↑	: Scrollt den Bildschirm und bewegt den Cursor mit
SHIFT ← →	: Bewegt den Cursor an den linken/rechten Rand
Backspace	: Löscht das Zeichen vor dem Cursor
TAB	: Der Cursor springt zur nächsten Ber Position
CONTROL RETURN	: Setzt den Cursor in die linke untere Ecke
SHIFT RETURN	: Schreibt CR
Clr/Home	: Setzt den Cursor in die obere linke Ecke
SHIFT Clr/Home	: Löscht den Bildschirm, setzt den Cursor nach oben links
ESC	: Löscht den Bildschirm ab der Cursorposition
Delete	: Löscht das Zeichen unterm Cursor, Rest rückt auf
SHIFT Delete	: Löscht die Zeile, in der der Cursor steht, Rest rückt auf
CONTROL Delete	: Löscht die Zeile ab der Cursorposition
Insert	: Fügt ein Space an der Cursorposition ein
SHIFT Insert	: Fügt eine Zeile ab der Cursorposition ein
CONTROL Insert	: Schaltet den Insert-Modus an (Autoinsert)
ALTERNATE Insert	: Schaltet den Insert-Modus aus
Undo	: Restauriert die Zeile (bis zum letzten RETURN)
Help	: Schreibt eine Zeile zurück, die mit
SHIFT Help	: gespeichert wurde
CONTROL 1 - 9	: Speichert den Bildschirminhalt in einen von zehn Buffern ab. Mit 1 - 9 sind die Tasten über der Alpha-Tastatur gemeint. Es wird nur der Teil unter der Registeranzeige gespeichert!
ALTERNATE 1 - 9	: Holt den Bildschirminhalt aus einem Buffer zurück
ALTERNATE Zahl	: Wenn Sie eine Zahl über den Zehnerblock eingeben, wird das entsprechende ASCII-Zeichen ausgegeben. Drücken Sie z. B. ALTERNATE 138 (dann ALTERNATE loslassen), wird e ausgegeben.
CONTROL ← →	: Damit können Sie durch den Cache blättern (entspricht den Pfeilen links neben der Registeranzeige) (siehe UNTRACE)
CONTROL Clr/Home	: Zeigt die aktuellen Register an (entspricht dem "Fuller" links von der Registeranzeige)
Funktionstaste	: Wählt eine der oben angezeigten Befehle an (siehe 5.2.)
CTRL+ALT+Delete	: Warmstart des Debuggers
CTR L+ALT+Rechtes SFT+Del	: Kaltstart des Debuggers (ist das gleiche wie der RESET-Taster. Mit zwei Resets schnell hintereinander wird der Debugger verlassen.)
CONTROL + P	: Setzt den PC an die aktuelle Adresse
CONTROL + B + 0-9/A-F	: Setzt einen Breakpoint an die aktuelle Adresse
CONTROL + G	: Startet das Programm ab der aktuellen Adresse (die aktuelle Adresse steht am Zeilenanfang)
SHIFT-SHIFT (beide)	: das laufende Programm wird unterbrochen und es wird in den Debugger zurückgekehrt. Der Debugger wird beendet und springt in den Assembler zurück. Das ist nur möglich, wenn der Debugger vom Assembler gestartet wurde.
CONTROL + Help	: Der Bildschirminhalt wird im DEGAS-Format abgespeichert.
CONTROL + SHIFT + Help	: Umschaltung Zehnerblock
Caps Lock	

normal und mit SHIFT

() / *
7 8 9 -
4 5 6 +
1 2 3 En-
0 . ter

Mit Caps Lock ohne SHIFT

A B C D
7 8 9 E
4 5 6 F
1 2 3 En-
0 . ter

Referenzkarte OMIKRON.Assembler

Die Tastaturbelegung des Assemblers

Backspace	: löscht das Zeichen links vom Cursor
Delete	: löscht das Zeichen unter dem Cursor
Insert	: fügt ein Leerzeichen an der Cursorposition ein
SHIFT Insert	: fügt eine Leerzeile ein
CONTROL Delete	: löscht die aktuelle Zeile
CONTROL Y	: löscht die aktuelle Zeile
CONTROL Insert	: schaltet die Editiermodi um (Überschr./Autoinsert I/II)
CLR/Home	: zum Anfang der Seite
SHIFT CLR/Home	: zum Ende der Seite
Tab	: zur nächsten Tabulatorposition
SHIFT Tab	: zur nächsten Tabulatorposition ohne einzurücken.
SHIFT Return	: zur nächsten Zeile und fügt eine Leerzeile ein.
CONTROL →	: zum Ende der Zeile
CONTROL ←	: zum Anfang der Zeile
CONTROL Pfeil hoch	: blättert eine Seite nach oben
CONTROL Pfeil runter	: blättert eine Seite nach unten
CONTROL CLR/Home	: zum Anfang des Sourcetexts
CONTROL Z	: zur Position der letzten Änderung
CONTROL J	: zum nächsten Fehler
CONTROL S	: zur nächsten ";x"-Marke
CONTROL SHIFT S	: zur vorhergehenden ";x"-Marke
ALTERNATE E	: zum Ende des Sourcetexts
ALTERNATE B	: zum Blockanfang
ALTERNATE K	: zum Blockende
ALTERNATE Z	: zur Zeile (akt. Z. wird für CTRL Z gemerkt)
ALTERNATE SHIFT Z	: zur Zeile o. erneute Abfrage des Labelnamens
ALTERNATE S	: zur nächsten nicht-tokenisierbaren Z. (A-Load)
ALTERNATE SHIFT S	: zur letzten nicht-tokenisierbaren Zeile
ALTERNATE J	: zum nächsten Fehler
ALTERNATE SHIFT J	: zum vorherigen Fehler
ALTERNATE 1 - 0	: zu einer Positionsmarkierung
ALTERNATE B	: zum Anfang des TEXT-Segments
ALTERNATE .	: zum Anfang des DATA-Segments
ALTERNATE #	: zum Anfang des Block-Storage-Segments (BSS)
CONTROL 1-0	: setzt eine Positionsmarkierung
CONTROL SHIFT 1-0	: löscht eine Positionsmarkierung
ALTERNATE 1 - 0	: zu einer Positionsmarkierung
CONTROL B	: markiert den Blockanfang (auch mit der Maus)
CONTROL K	: markiert das Blockende (auch mit der Maus)
ALTERNATE B	: zum Blockanfang
ALTERNATE K	: zum Blockende
CONTROL H	: löscht die Blockmarkierungen (Hide Block)
ALTERNATE Y	: löscht den Block
CONTROL D	: dupliziert die Zeile, in der der Cursor steht.
CONTROL F	: sucht ein Symbol (nur zu Symbolaufrufen)
CONTROL L	: sucht einen ASCII-Text
CONTROL E	: sucht und ersetzt einen ASCII-Text
ALTERNATE F	: setzt die Symbolsuche fort
ALTERNATE L	: setzt die ASCII-Suche fort
CONTROL W	: Zahlenbasis wandeln (Dec/Hex/Bin/"ASC"/"ASC")
CONTROL +	: Zeichensatz ausgeben (Auswahl mit Maus)
ALTERNATE SHIFT Help	: Speichert den Bildschirminhalt im DEGAS-Format

Pseudopcodes des Assemblers

~DATE	27	ergibt das Datum im TOS-Format
~TIME	27	ergibt die Uhrzeit im TOS-Format
~RSCOUNT	27	ergibt den akt. Stand des RS-Counters
RASE (DC.(BIWIL)) (AO-A7)		stellt einen Adressierungsoffset ein
TEXT (DATA) (BSS) Label	33	
SECTION) BSS (BSS)	30	Beginn des Block-Storage-Segments
ONOP Offset, Justierung	34	PC wird durch Justierung teilbar
SECTION) DATA (.DATA)	30	Beginn des DATA-Segments
DC.(BIWIL) Wert, Wert (DC.(BIWIL))	28	definiert konstanten
DS.(BIWIL) Wert (Füllwert)		reserviert Speicher
(DS.(BIWIL) BLK.(BIWIL) I		(Größe in BIWIL = Wert)
DSBSS.(BIWIL) (DCB.(BIWIL))	28	
DXE.BI 'String'	28	definiert Stringkonstante mit fester Länge
DXSET Länge (Füllwert)	28	setzt Länge der Stringkonstante
ELSE	36	wird übersetzt, wenn Bedingung nicht erfüllt
END (END)	31	Ende des Sourcecodes
ENDC (ENDIF)	35	Ende einer IF-Struktur (bedingte Assembl.)
ENDR	35	Ende einer REPEAT-Schleife
EVEN (EVEN) ALIGN.W)	28	begründet die nächste Adresse
FAIL {text}	36	gibt Fehlertext aus (bedingte Assemblierung)
IBYTES 'Filename'		
[Anzahl] (Position) (IBIBIN)	32	lädt ein File von Diskette nach
IF(NE) Bedingung	36	Assemblieren, falls Bedingung erfüllt
IFD Label	36	Assemblieren, falls Label definiert
IFEQ Bedingung	36	Assemblieren, falls Bedingung nicht erfüllt
IFGE Ausdruck	36	Assemblieren, falls Ausdruck >= 0
IFGT Ausdruck	36	Assemblieren, falls Ausdruck > 0
IFLE Ausdruck	36	Assemblieren, falls Ausdruck <= 0
IFLT Ausdruck	36	Assemblieren, falls Ausdruck < 0
IFND Label	36	Assemblieren, falls Label nicht definiert
ILLEGAL	35	fügt \$4AFC ein, löst illegal instruction aus
OPT (DIWISIMICN) (+) (-) (..)	31	stellt Assembleroptionen ein
ORG Adresse[, ^]	35	Stellt Adresse für absolute Assembl. ein
OUTPUT	32	Stellt Namen ein, unter dem gespeichert wird
PATH	32	Pfadname für IBYTES
REPT Anzahl	35	Beginn einer REPEAT-Struktur
RS.(BIWIL)	29	reserviert Speicher
RSRESET	29	Setzt RS-Counter zurück
RSET Wert	29	Setzt den RS-Counter
RSEVEN	30	Begründet den RS-Counter
RSBSS	30	Addiert RS-Counter zur BSS-Länge, RS=0
Symbol EQU Wert (EQU=1==1..)	31	Ordnet einem Label einen Wert zu
Symbol REG Registerliste	31	Definiert Registerliste für movem
SECTION) TEXT (.TEXT)	30	Beginn des Textsegments

Die Debuggerbefehle

) ASCII-Daten	65	ASC-Präfix steht vor ASC-Dump
,Term[,Term]	64	Schreibt Daten in Speicher (byteweise)
,WIL)Term[,Term]	64	Schreibt Daten wort- oder langwortweise
/ Hex-Daten	64	Schreibt Hex-Daten (Disassemble)
?Term[,Term]	64	Berechnet Term und gibt ihn aus
ASCIND [Von,Bis],String	61	Sucht nach Strings in Disass.-Listings
ASCII [Von][Zu][Zellen][Bis]	67	Zeigt Speicherbereich an (ASC)
(BREAKPOINTSIB) [(Nummer[=Adresse[,*=Wert]Anzahl		
?Bedingung)] [(K[Nummer])]	56	Setzt/Löscht/Zeigt Breakpoints
BSSCLEAR	76	Löscht das BSS
CHECKSUMME [Adr [,Prüfsum[,Anzahl]]]	75	Berechnet eine Prüfsumme
CLR [Von][Bis]	72	Löscht einen Speicherbereich
CLRCACHE	71	Löscht Cache (Register/Befehlsspeicher)
CLS	172	Löscht den Bildschirm
COMPARE Von,Bis,Adresse	72	Vergleicht zwei Speicherbereiche
CONTINUE	62	Setzt Suche fort (ASCIND, FIND, HUNT)
(MOVEICOPY) Von,Bis,Nach	77	Kopiert einen Speicherbereich

CR [Anzahl]	65	gibt (Anzahl) Leerzellen aus
CURSOR Nummer	77	Stellt Aussehen des Cursors ein
DIRECTORY [Pfad][Suchmaske]	156	Inhaltsverzeichnis, stellt Pfadnamen ein
DISASSEMBLE [Von][Zu][Zellen][Bis]	57	Disassembliert, mit Hex-Code ohne Labels
DOICALL [Adresse][Breakpoint]	67	Führt ein Unterprogramm aus
(DUMPIMEMORY)[(BIWIL)] [Von][Zu][Zellen][Bis]		
	66	Zeigt einen Speicherbereich an (Hex/ASC)
(ERASEKILL) [Filename]	173	Löscht ein File
EXITQUITSYSTEM	60	Verläßt den Debugger
FATTRIBUT [Name][Attribut]	174	Setzt Fileattribute
FCLOSE	165	Schließt eine Datei (für FILE)
FILEBefehl	165	Ausgabe des Folgebefehls auf File
FILL Von,Bis,Wert[,Wert]	71	Füllt einen Speicherbereich
FIND [Von,Bis],Term	61	Durchsucht einen Speicherbereich
FOPEN Filename	165	Öffnet ein File (für FILE)
FORMAT [(DIS)SJE[,Laufwerk]	174	Formatiert eine Diskette
FREE [Laufwerk]	173	Ermittelt freien Speicher auf Laufwerk
GETCACHE	71	Holt Registerwerte aus Cache
GETREGISTER [Adresse]	75	Holt Registerwerte aus Speicher
GO [Adresse][Breakpoint]	60	Startet ein Programm ab Adresse/PC
HUNT [Von,Bis],Term	61	Durchsucht einen Speicherbereich (.w)
IF [Register Bedingung] Term	68	Stellt Abbruchbedingung ein
INFO	62	Gibt Start von TEXT-, DATA- und BSS aus
INITREGISTER	76	Setzt alle Register auf null
KEY [(ICIA)F Nummer=Text)]	72	Belegt eine Funktionstaste
(ERASEKILL) [Filename]	173	Löscht ein File
((SIL)ET)- (ALLVariable)=Term	66	Setzt ein(alles) Register
LEXECUTE [Filename][, "Commandline"]	55	Lädt ein Programm zum Debuggen (Reloziert)
LINF	65	gibt 79 mal "-" aus
LIST [Von][Zu][Zellen][Bis]	58	Disassembliert, mit Labels ohne Hex-Code
LOAD [Filename][Adresse]	155	Lädt ein File in den Speicher
(DUMPIMEMORY)[(BIWIL)] [Von][Zu][Zellen][Bis]		
	66	Zeigt einen Speicherbereich an (Hex/ASC)
MOUSEJOFF	173	Schaltet die Maus aus
MOUSEON	173	Schaltet die Maus ein
(MOVEICOPY) Von,Bis,Nach	77	Kopiert einen Speicherbereich
MKDIRECTORY Filename	173	Richtet ein Unterverzeichnis ein
NAME Altname,Neuname	174	Benennt ein File um
OBSERVE [Trap[, (? -)IListe]]	70	Fängt bestimmte Traps ab
! Opcode	64	Präfix Opcode (Ausgabe von LIST)
PRNBefehl	65	Ausgabe des Folgebefehls auf Drucker
(READTRACKIRTRACK) [Track[,Seite[,Adresse[,Laufwerk]]]]		
	79	Liest einen kompletten Track
(RSIREADSECTOR) [Track[,Sektor[,Seite[,Adresse		
[,Laufwerk]]]]	178	Liest Sektoren (XBIOS 8)
RESET [ALL][, Vektoren]	75	Die Systemvektoren gesetzt (auf Debugger)
RESIDENT	82	Verläßt den Debugger (bleibt resident)
RMDIRECTORY [Filename]	173	Löscht ein Unterverzeichnis (muß leer sein)
RWABS Rwf[,Buf[,Anzahl,Recno[,Laufwerk]	178	Schreibt (Liest) Sektoren (BIOS 4)
SAVE [Dateiname][,Anfadr[,Endadr]	163	Speichert einen Speicherbereich auf Disk
SAVE [HJR]	177	Speichert Einstellungen
SCROLLOFF/SCROLLON [Fangradius]	77	Schaltet Auto-Scrolling ein(aus)
SHOW (Adr [(BIWIL)[Anzahl])[(,Num)	66	Zeigt Speicherbereich ständig an
"String")]	66	String ausgeben (PRN oder FILE)
((SIL)ET)- (ALLVariable)=Term	66	Setzt ein(alles) Register
SWITCH	76	Schaltet Auflösung um
SYMBOLTABLE [Von][Zu][Zellen][Bis]	58	Gibt Symboltabelle aus
SYNC50 / SYNC60	77	Stellt Bildfrequenz ein
]Term	64	Schreibt Daten (ohne Neuausgabe)
TRACE [Anzahl]	63	Führt einen(mehrere) Befehle aus (F1)
TYPE [Dateiname]	178	Anzeigen einer Datei (ASC)
UNTRACE [Adresse[,Anzahl]]	67	Führt ein Programm bis zur Abbruchbed. aus
(WSIWRITESEKTOR) [Track[,Sektor[,Seite[,Adresse		
[,Laufwerk]]]]	179	Schreibt Sektoren (XBIOS 8)
!Befehl	79	Führt einen Befehl sofort aus