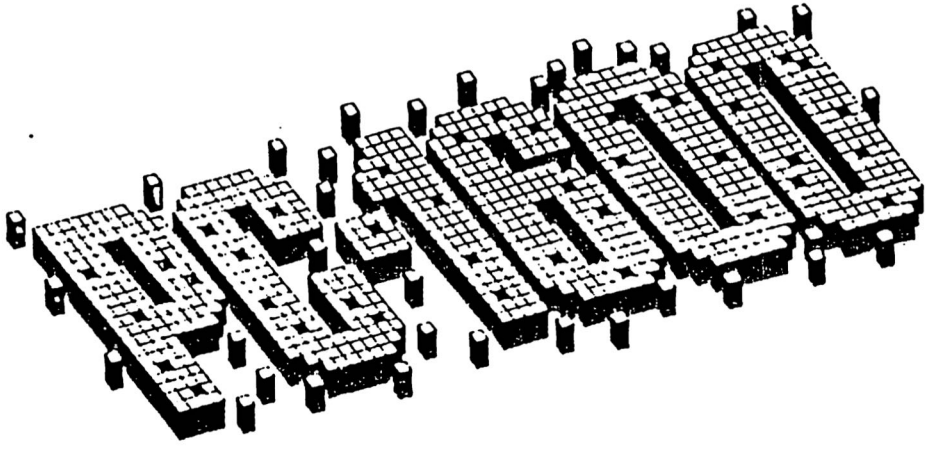
SHARP

DEBUGGER



Klaus Ditze, Weilerswist

Copyright

Einschränkung der Gewährleistung

DEBUGGER wurde mit größtmöglicher Sorgfalt erstellt und geprüft. Es wird keine Garantie, weder in Bezug auf diese Anleitung, noch in Bezug auf die in dieser Anleitung beschriebene Software, ihre Qualität, Durchführbarkeit oder Verwendbarkeit für einen bestimmten Zweck übernommen. Weiterhin wird in keinem Fall für direkte, indirekte, verursachte oder gefolgte Schäden, die entweder aus unsachgemäßer Bedienung oder aus irgendwelchen Fehlern an der Software resultieren, gehaftet. Da sich Fehler trotz aller Bemühungen nie vollständig vermeiden lassen, sind wir für Hinweise jederzeit dankbar.

Funktionsbeschreibung

Mit dem PC-1600 Debugger können Sie Maschinenprogramme disassemblieren, ausdrucken und mit Single Step oder Breakpoint testen.
Der Debugger ist besonders als Ergänzung zum PC-1600 Assembler[1] geeignet, da er dessen Symboltabellen lesen kann und symbolisches Debuggen ermöglicht.

Senden Sie unbedingt die beigefügte Kundenkarte an uns ein, damit wir Sie bei eventuellen Programmänderungen informieren können.
Eine eventuell notwendige Erweiterung oder Fehlerberichtigung dieser Anleitung ist auf der Programmdiskette unter dem Namen 'DB-NEU' abgespeichert.
Laden Sie dieses Programm im RUN-Modus mit LOAD "X:DB-NEU",R. Alles weitere entnehmen Sie dann der Rechneranzeige.

Bevor Sie mit Ihrer Arbeit beginnen, fertigen Sie unbedingt eine Arbeitskopie Ihrer *DEBUGGER* Diskette an. Arbeiten Sie unbedingt auf keinem Fall mit der Orginaldiskette. Diese könnte beschädigt werden.

Start

Damit der Debugger mit Ihren eigenen Programmen nicht in Konflikt gerät, wurde er völlig relokatibel ausgelegt. So können Sie ihn mit

BLOAD "X:DB.REL",*#Bank,Adresse*

an jede beliebige RAM-Adresse laden.
Am besten belegen Sie eine Funktionstaste mit dem Aufruf

CALL *#Bank,Adresse*@

,um den Debugger bequem zu starten.

Der Debugger steht, wenn er ohne Adreßangabe geladen wurde, ab der Adresse &C0C5.
Zum Speichern von Symboltabellen benötigt er einen weiteren Speicherbereich, den Sie mit dem CONFIG-Menuepunkt Ihren Erfordernissen anpassen können.
Außerdem benötigt das Programm die BASIC-Variablen E$ - Z$ als Arbeitsspeicher.

Weiterhin wird der Vektor des RST 38-Befehls verwendet. Wenn dieser schon belegt ist, erscheint die Meldung

Vector &38 already in use

Geben Sie in diesem Fall POKE &F0D7,&C9 ein.

[1] Zu beziehen über Ihren Fachhändler oder der Firma Klaus Ditze, Weilerswist

auptmenue

Nach einer kurzen Copyright-Meldung erscheint das Hauptmenue.
Die einzelnen Menuepunkte erreichen Sie durch Drücken der jeweils angegebenen Taste:

P-Prog :	Programm disassembliert anzeigen und starten
D-Data :	Daten als Hexdump anzeigen und ändern
S-Step :	Schrittweite des Single Step setzen
I-Info :	Informationen über das Programm
R-Bload:	Maschinenprogramm laden
W-Bsave:	Maschinenprogramm speichern
T-Table:	Symboltabelle laden
E-Clear:	Symboltabelle löschen
B-Break:	Breakpoint-Adresse setzen
L-Lprnt:	Programm disassembliert ausdrucken
C-Confg:	Voreinstellungen ändern und speichern
Q-Quit :	Programm verlassen

Weiterhin zeigt Ihnen die <RCL>-Taste die Seriennummer an. Bitte geben Sie diese immer an, wenn Sie Fragen zum Programm haben.

Mit der <OFF>-Taste können Sie den Rechner jederzeit abschalten, und mit der <MODE>-Taste verlassen Sie den Debugger.

Bei der Eingabe von Adressen o.ä. lassen sich die Tasten <Cursor links>, <Cursor rechts>, CL, BS, SHIFT-INS und SHIFT-DEL wie gewohnt zur Korrektor benutzen. Die CTRL-Taste schaltet zwischen Einfüge- und Überschreib-Modus um. Die MODE-Taste bricht die Eingabe ab.

Data

Zunächst müssen Sie die gewünschte Adresse eingeben. Falls Sie mit der vorgeschlagenen Adresse einverstanden sind, genügt es, <ENTER> zu drücken.
Wenn die vorgegebene Adresse den Wert einer symbolischen Marke hat, wird diese, durch ein "="-Zeichen getrennt, mit angezeigt.

Als Adresse ist eine bis zu 4-stellige Hexadezimalzahl anzugeben, die von einem Komma und der Speicher-Bank gefolgt wird.
Als Bank ist möglich:

0-7	: Bank 0-7
3B	: Verstecktes BASIC-ROM
4B	: Japan-ROM
20-27	,
30-37	: Extra-Bankumschaltung für große Speichermodule in Slot 2
X:,Y	: Diskette X: bzw. Y:, hierbei wird die Diskette als virtueller Speicher mit dem Adreßbereich 0000-FFFF angesehen. Wenn die Diskette nicht schreibgeschützt ist, werden Änderungen der Daten beim Wechsel aus dem aktuellen Sektor auf Diskette geschrieben.

Anstelle der Hexadezimalzahl für die Adresse können Sie auch den Namen einer symbolische Marke angeben, falls diese in der geladenen Symboltabelle existiert.

Nun wird ein 4-stelliges Hexdump mit den dazugehörigen Adressen, Bank und ASCII-Codes angezeigt, in dem Sie mit den Pfeiltasten herumwandern und die Datenwerte ändern können.

Weitere Tastenfunktionen:

Doppelpfeil	: Umschaltung von ASCII auf Hex und umgekehrt
RCL	: Neue Adresse
MODE	: Zurück ins Hauptmenue

Prog

Dieser Menuepunkt dient zum Testen von Programmen. Daher sehen Sie zunächst den Inhalt aller CPU-Register. An rechten Rand der Anzeige stehen der Zustand des Carry- und Zero-Flag, sowie das I-Register und die Speicherbank. Mit den Pfeiltasten (recht und links) können Sie die gewünschte Position erreichen, um die Register zu ändern. Die Register lassen sich auch über bestimmte DEF-Tastenkombinationen anwählen:

DEF-B : BCreg	DEF-D : DEreg	DEF-H : HLreg
DEF-C : BC'reg	DEF-E : DE'reg	DEF-L : HL'reg
DEF-A : AFreg	DEF-X : IXreg	DEF-S : SPreg
DEF-F : AF'reg	DEF-Y : IYreg	DEF-P : PCreg

Der Stackpointer steht zu Beginn auf &F560 und sollte beim Testen möglichst im Stack-Bereich bis &F500 bleiben.
Wenn durch das zu testende Programm die Speicherbank gesetzt wird, sollten Sie diese während des Testvorgangs nicht mehr selbst ändern, da dies sonst zu falschen Ergebnissen führen kann.

Die Pfeiltasten (auf und ab) dienen zum vor- und rückwärtsblättern der Disassembler-Anzeige.

Single Step und Breakpoint werden durch ein besonderes Simulationsverfahren verwirklicht, das auch auf ROM-Bereichen funktioniert und dennoch die Interrupt-Steuerung nicht beeinflußt.

Da weder Breakpoint noch Single Step in Echtzeit ablaufen, sollten keine zeitkritischen Programme, wie z.B. Disketten- oder Druckersteuerung, getestet werden. Dies könnte zu Defekten in der Hardware führen.

Testfunktionen:

DEF-G : Start ohne Single Step
Abbruch beim Erreichen des Breakpoints oder am abschließenden RET, d.h. wenn der Stackpointer >&F56Ø ist

ENTER : Single Step
Wie DEF-G, aber höchstens so viele Befehle wie mit STEP eingestellt

SPACE : Befehle überspringen
Wie ENTER, aber die Befehle werden nicht wirklich ausgeführt

DEF-SPACE: Unterprogramm ausführen
Wenn die aktuelle Adresse einen Unterprogramm-Aufruf mit CALL oder RST enthält, wird dieses Unterprogramm komplett durchlaufen und am Befehl hinter dem Aufruf wieder angehalten, sonst wie DEF-G

Weitere Tastenfunktionen:

Doppelpfeil: Umschaltung zwischen Registeranzeige und dem aktuellen Befehl (+ den 3 folgenden) und umgekehrt

RCL : Neue Adresse

MODE : Zurück ins Hauptmenue

Ein kleiner Gag:
Gehen Sie einmal zur Adresse &Ø112 (das ist die CLS-Routine) und drücken Sie DEF-G. Damit haben Sie auch eine Vorstellung, wie lang die Simulation der Befehle für Single Step und Breakpoint dauert.

Break

Setzen des Breakpoints
Natürlich können Sie auch hier symbolische Namen verwenden.

Step

Setzen der Schrittweite (=Anzahl der Befehle) eines Single Steps.

Bload

Wenn Sie Maschinenprogramme mit diesem Menuepunkt laden, teilt Ihnen der Debugger, nachdem Sie den Dateinamen eingegeben haben, zunächst die Startadresse, Bank, Endadresse und Autostart-Adresse mit. Jetzt können Sie, wenn nötig, erst noch die Ladeadresse ändern. Ein eventuell vorhandener Autostart wird ignoriert.

Bsave

Wie vom entsprechenden BASIC-Befehl gewohnt, geben Sie hier Dateiname, Startadresse mit Bank, Endadresse und Autostart an.
Wenn Sie keinen Autostart wünschen, geben Sie hierfür &FFFF ein.

Table

Mit diesem Menuepunkt können Sie die Symboltabelle für ein zu testendes Programm laden. Geben Sie hierzu den Dateinamen an.
Typischerweise wurde diese Symboltabelle durch den PC-1600 Assembler erzeugt. Falls nicht, muß folgendes Dateiformat eingehalten werden, sonst erscheint ERROR 157:

ASCII-Datei mit CTRL-Z am Datei-Ende (Code &1A)
Alphabetisch nach Symbolnamen geordnete Zeilen mit Zeilenend-Code CR + LF (&0D,&0A) und folgendem Aufbau:

max. 4-stellige Hexadezimalzahl
Space (Code &20)
max. 6-stelliger Symbolname
z.B.: "80C5 Start"

Clear

Löscht die Symboltabelle im Speicher, wenn Sie sie momentan nicht benötigen

Lprnt

Zum Ausdrucken müssen Sie lediglich Start- und Endadresse angeben.
Das Programm erkennt selbst, ob der Plotter oder das Centronics-Interface angeschlossen ist. Die in BASIC definierten Druckparamter werden eingehalten. Achten Sie hierbei besonders auf den richtigen Zeilenend-Code (mit PCONSOLE "LPTx",,Code).

Info

Dieser Menuepunkt zeigt Ihnen u.a. die Versions-Nummer des Programms.

Config

Hier können Sie die Start- und Endadresse des Symbolspeichers definieren. Beide Adressen müssen Vielfache von 16 sein, d.h. die Hexadezimalzahl muß mit einer Null enden. Pro Symbol werden 12 Byte Speicher benötigt. Wenn sich beim Laden der Symboltabelle dieser Bereich als zu klein erweist, erscheint ERROR 30.

Beispiel:
Wenn Sie die Bank 1 eines RAM-Moduls "S1:" für den Debugger verwenden wollen, könnte die Speicheraufteilung folgendermaßen aussehen:

```
Debugger Anfang      : &8000
Symbolspeicher Start : &A000
               Ende  : &C000
```

Damit ist Speicher für (&2000/12)=682 Symbole reserviert.

Der Symbolspeicher liegt immer auf der gleichen Bank wie der Debugger selbst.

Die nächste Abfrage legt fest, ob bei Eingaben der Einfüge-Modus zunächst eingeschaltet ist oder nicht.

Schließlich gibt es die Möglichkeit, das konfigurierte Programm unter dem Namen "X:DBP.REL" abzuspeichern.

Beispiel

Folgendes kleine Maschinenprogramm wurde mit dem Assembler übersetzt. Es schreibt 'HALLO' in die Anzeige.

Quelltext für PC-1600 Assembler.

```
'Hallo
'
'Demoprogramm für den
'PC-1600 ASSEMBLER
'
'
EQU START &F900 'Das Programm wird in die Standardvariablen geladen
'
EQU CLS &0112
EQU HOME &0109
EQU PRINT &00EB
EQU CR &0D
'
ORG START
AUTO START
'
CALL CLS
CALL HOME
LD DE,TEXT
LD A,CR
CALL PRINT
RET
'
TEXT:
DEFB "**  Hallo  **"
DEFB CR
'
END
```

Bei der Assemblierung wird folgende Symboltabelle erzeugt:

```
0112 CLS
000D CR
0109 HOME
00EB PRINT
F900 START
F90F TEXT
```

Die Ausgabe auf dem Drucker sieht, wenn 'HALLO.BIN'[2] sowie 'HALLO.SYM'[3] geladen sind, wie folgt aus:

```
F900=START  ,0  :  CD 12 01   -     CALL 0112=CLS
F903        ,0  :  CD 09 01   -     CALL 0109=HOME
F906        ,0  :  11 0F F9     .   LD   DE,F90F=TEXT
F909        ,0  :  3E 0D        >   LD   A,0D
F90B        ,0  :  CD EB 00   =ó    CALL 00EB=PRINT
F90E        ,0  :  C9           ?   RET
F90F=TEXT   ,0  :  2A 2A 20   **    LD   HL,(202A)
F912        ,0  :  20 48        H   JR   NZ,>F95C
F914        ,0  :  61           a   LD   H,C
F915        ,0  :  6C           l   LD   L,H
F916        ,0  :  6C           l   LD   L,H
F917        ,0  :  6F           o   LD   L,A
F918        ,0  :  20 20            JR   NZ,>F93A
F91A        ,0  :  2A 2A 0D   **    LD   HL,(0D2A)
F91D        ,0  :  00               NOP
```

Disketteninhalt

Auf der mitgelieferten Diskette befinden sich folgende Dateien:

- DB.REL	Debugger, relokatibel
- HALLO.ASM	Quelltext für HALLO.BIN
HALLO.SYM	Bei der Assemblierung[4] von HALLO.ASM erzeugtes Symbolfile
HALLO.BIN	Objektfile, Adreßbereich: F900-F91D

Technische Daten

Softwareprogramm:	*DEBUGGER* für SHARP PC-1600
Adreßlage:	C0C5-DD26 (Version 1.00)
Programmlänge:	7265 Bytes (Version 1.00)
Programmiersprache:	Z80 Assembler
Programmautor:	David von Oheimb
(c) Copyright:	Klaus Ditze, D-5354 Weilerswist

[2] Im Hauptmenue mit R-BLoad laden
[3] Im Hauptmenue mit T-Tabelle laden
[4] Mit dem PC-1600 Assembler

Literaturhinweise

- David von Oheimb
  Das Systemhandbuch für den PC-1600
  ISBN 3-925641-08-4
  bei: Klaus Ditze, D-5354 Weilerswist

- Rodnay Zaks
  Programmierung des Z80
  bei: Sybex, Düsseldorf

Firma Klaus Ditze
Nikolaus-Otto-Str. 6
D-5354 Weilerswist