

SHARP PC-1600 DEBUGGER von Klaus Ditze

Start

Damit der Debugger mit ihren eigenen Programmen nicht in Konflikt gerät, wurde er völlig relocatibel ausgelegt. So können Sie ihn mit `BLOAD "S2:DEBUGGER.BIN",#Bank,Adresse` an jede beliebige RAM-Adresse laden. Der DEBUGGER steht wenn er ohne Adressangaben geladen wurde ganz vorne im RAM bei Adresse `&80C5` und kann dann mit `BLOAD "S2:DEBUGGER.BIN"` geladen werden. Ohne Ramerweiterung sollte er mit `BLOAD "S2:DEBUGGER.BIN",#0,&C0C5` geladen und dann mit `CALL &C0C5` gestartet werden.

Hauptmenue

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

P-Prog:	Programm disassembiert 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 disassembiert ausdrucken
C-Config:	Voreinstellungen ändern und speichern
Q-Quit:	Programm verlassen

Weiterhin zeigt 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 Korrektur 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 symbolischen 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 Hauptmenü

Prog

Dieser Menüpunkt 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 (rechts und links) können Sie die gewünschte Position erreichen, um die Register zu ändern. Die Register lassen sich auch über bestimmte DEF-Tastenkombinationen auswählen:

DEF-B: BCreg DEF-C : BC 'reg DEF-A: AFreg DEF-F : AF'reg

DEF-D: DEreg DEF-E : DE 'reg DEF-X: IXreg DEF-Y: IYreg

DEF-H: HLreg DEF-L : HL ' reg DEF-S: SPreg 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 rückwärts blättern der Disassembleranzeige.

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 Hardwareführen.

Testfunktionen:

- DEF-G . : Start ohne Single Step. Abbruch beim Erreichen des Breakpoints oder am abschließenden RET. d.h. wenn der Stackpointer >&F560 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 Hauptmenü

Ein kleiner Gag: Gehen Sie einmal zur Adresse &0112 (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.

Load Wenn Sie Maschinenprogramme mit diesem Menüpunkt 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.

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 Menüpunkt können Sie die Symboltabelle für ein zu testendes Programm laden. Geben Sie hierzu den Dateinamen an. Typischerweise wurde diese Symboltabelle durch den PC-16U0 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 Zellen mit Zeilenend-Code CR + LF (&0D,&0A) und folgendem Aufbau:

max. 4-stellige Hexadezimalzahl Space (Code &20)

max. 6-stelliger Symbolname z.B.: &C0C5 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 Druckparameter werden eingehalten. Achten Sie hierbei besonders auf den richtigen Zeilenend-Code (mit PCONSOLE "LPTx", Code).

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    ' Variable CLS wird mit Wert &0112 gesetzt
                  ( Unterprogrammroutine Bildschirm löschen)
EQU HOME &0109   ' Variable HOME wird mit Wert &0109 gesetzt
                  ( Unterprogrammroutine Cursor links Oben positionieren)
EQU PRINT &00EB  ' Variable PRINT wird mit Wert &00EB gesetzt
                  ( Unterprogrammroutine Zeige Zeichen ab DE-Register bis
                  Endzeichen im A-Register)
EQU CR &0D       ' Variable CR wird mit Wert &0D ( 13 ) gesetzt.
,
ORG START        ' Startadresse Maschinenprogramm
AUTO START       ' Das Maschinenprogramm soll nach BLOAD automatisch
                  Starten
,
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