

IDAS

Ein Interaktiver Dialogassembler für KC 87

Der IDAS ist ein Dienstprogramm zur dialogorientierten Erarbeitung von Programmen in der Assemblersprache SYPS K 1520. Er vereinigt in sich Editier-, Übersetzer- und Interpreterfunktionen. Besonders geeignet ist er dann, wenn Befehlsfolgen und kurze Programme (z. B. Unterprogramme für den BASIC-Interpreter) rationell aufgestellt und getestet werden sollen.

Mit IDAS hat der Anwender die Möglichkeit, Programme auf Assemblerniveau zu schreiben. IDAS erzeugt daraus Maschinencode. Das Besondere am IDAS besteht in seiner einfachen Bedienung und der Möglichkeit, das Programm abschnittsweise abarbeiten zu lassen. Damit werden gute Voraussetzungen für einen effektiven Programmtest geschaffen. Der IDAS umfaßt die Funktionen eines Editors (Programmaufbereitung), Assemblers (Übersetzung) und Laders (Installieren und Starten des Programms).

Speicherbelegung

| | |
|------------------|--|
| Dateiname | IDAS.COM |
| Laden in | OS |
| Programmstandort | 0400h-1BFFh (RAM-Version) 0C000h-0E7FFh (ROM-Version) |
| OS-Kommando | IDAS |

Der Speicher von 300h-400h wird als Arbeitsspeicher genutzt.

Es gibt zwei ROM-Versionen: Da der IDAS nur ~ 6KByte Speicher belegt, sind im Modul noch 4K ROM frei. Mein IDAS-Modul meldet sich mit „IDAS :“ und enthält im zusätzlichen Speicher den ZM 2.0 (Z9001 MONITOR V2.0 (ROM) 1985) und die EPROM-Software für das Programmier-Modul.

Der IDAS bei U. Zander meldet sich mit „INTERPRETING DIALOG-ASSEMBLER“ und enthält zusätzlich den ZM (Z9001 MONITOR CENT. 1986), RENEW und QUICK,NORMAL,LOAD. RENEW stellt ein versehentlich mit NEW gelöscht Basic-Programm wieder her. Die Kassettenroutinen QUICK,NORMAL,LOAD erfordern einen speziellen Adapter am User-Port. Sie funktionieren nicht mit dem normalen Kassetteninterface!

Auch von der RAM-Version gibt es verschiedene Fassungen (s.u.).

Anleitung

Nach Start erscheint die Meldung 'IDAS: '; mit Eingabe des Buchstabens 'N' erfolgt ein Neustart, jede andere Taste bewirkt einen Restart. Eine Zuweisung der Speicherbereiche fuer Quelltext (TB) und Maschinencode (MC) ist mit dem Kommando **ASGN** möglich.

Standard: MC 1C00H - 1FFFFH und TB 2000H - ...

Bei Bereichsüberschreitung erfolgt eine Fehlermeldung (+TB+ / +MC+).

IDAS wird durch das Kommando **CM** (Call Monitor) verlassen.

Im Kommando-Mode werden die eingegebenen Befehle sofort ausgeführt. Die Befehle DI, IM0, IM1 sind hier verboten, da sie die Interrupt-Bearbeitung der CPU beeinflussen und dadurch die Tastaturabfrage blockieren. Danach hilft nur noch RESET. Es ist zu beachten, dass IDAS eventuell vorhandenen Quelltext übersetzt, bevor er einen Befehl im Kommandomode ausführt. Dadurch kann auch im Kommandomode auf Marken des Quelltextes Bezug genommen werden.

Im Editmode erfolgt die Eingabe des Quelltextes in der Form:

```
Zeilenr. Marke: Befehl
```

Eine Marke kann stehen, muss aber nicht. Kommentare sind nicht möglich.

Verarbeitet wird ROBOTRON-Code für U880.

Gegenüberstellung der Schreibweise einiger Beispielbefehle:

| ROBOTRON | Zilog | ROBOTRON | Zilog |
|------------|---------------|----------|-------------|
| M1: LD M,A | M1: LD (HL),A | OUT 80H | OUT (80H),A |
| JR M1-# | JR M1 | IN 3CH | IN A,(3CH) |
| JMP M1 | JP M1 | CMP 0 | CP 0 |
| JRC M1-# | JR C,M1 | ADD B | ADD A,B |
| JPZ M1 | JP Z,M1 | JMP M | JP (HL) |
| CANZ M1 | CALL NZ,M1 | RNC | RET NC |
| DJNZ M1-# | DJNZ M1 | | |

Steueranweisungen für den Assembler

| | |
|--------------------|---|
| ORG adresse | Anfangsadresse ; muss im MC-Bereich liegen |
| konstante:EQU wert | Wertzuweisung zu einem Symbolnamen |
| BER anzahl | Bereich freihalten (wird nicht gelöscht) |
| DB byte | Byte definieren |
| DB 'text' | Text definieren |
| DA adr | Adresse (Doppelbyte) definieren |
| END | Ende des Quelltextes ; danach dürfen keine Befehle mehr stehen, sonst Absturz möglich |

Zahlenwerte können dezimal (ohne Kennzeichnung) oder hexadezimal (mit nachgestelltem 'H') angegeben werden. Zahlen müssen mit einer Ziffer beginnen, d.h. Hexadezimalzahlen größer 9FH benötigen eine Vornull (z.B.: 0CBH). Die Verwendung von ASCII-Zeichen ist möglich, wenn diese in Hochkommas eingeschlossen werden.

Bsp.: LD A,75 und LD A,4BH und LD A,'K' sind identisch.

Kommandos

| | |
|------|--------------------------------|
| ASGN | Speicherplatzzuweisung (s.o.) |
| CM | Rückkehr ins OS (call monitor) |

| | |
|--------------------|--|
| LIST marke | Anzeige Quelltext ab Marke |
| LIST zeile1,zeile2 | Anzeige Quelltext ab Zeile1 bis Zeile2 |
| STEP abstand | neuer Abstand der Zeilennummern |
| KILL | Löschen des gesamten Quelltextes |
| KILL zeile1,zeile2 | Löschen Quelltext von Zeile1 bis Zeile2 |
| RUN | ganzes Programm übersetzen und starten |
| RUN zeile1 | Programm ab Zeile1 übersetzen und starten |
| RUN zeile1,zeile2 | wie RUN, aber nur von Zeile1 bis Zeile2 |
| ADDR markenname | Ausgabe der Markenadresse |
| PUT konstante | Ausgabe des Wertes der Konstanten |
| DR | Anzeige Register (display register) |
| DS | Speicher anzeigen / ändern (Ende mit 'STOP'-Taste) |
| READ | Eingabe von Quelltext (an den Anfang) durch 'DEVICE (T/C)' T=Tape oder C=Console zuweisen. READ von Console ermöglicht Eingabe ohne Zeilennummern. Diese werden automatisch erzeugt. |
| READ zeile1 | Einfügen von Quelltext ab Zeile1 |
| READ marke | Einfügen von Quelltext ab Marke |
| WRIT | Ausgabe von Quelltext durch 'DEVICE (T/C)' T=Tape oder C=Console zuweisen. |
| WRIT zeile1 | |
| WRIT marke | |
| MCSV | MC-SAVE. Maschinencode entsprechend der bei ASGN zugewiesenen Adressen auf Kassette schreiben. |
| MCSV adr1,adr2 | Speicherabzug von Adresse1 bis Adresse2 auf Kassette schreiben. |

Fehlermeldungen

| | |
|-------|------------------------------------|
| +ILO+ | unzulässige Operation |
| +IO+ | Ein-/Ausgabefehler |
| +LNR+ | unzulässige Zeilennummer |
| +MDS+ | mehrfach definiertes Symbol |
| +MNE+ | falsche Syntax ; unbekannter Code |
| +NDS+ | nicht definiertes Symbol |
| +OPD+ | Operandenfehler bzw. Operand fehlt |
| +SYM+ | Symbolfehler |

Ein beliebter Fehler: in der Eingabezeile sind Control-Zeichen enthalten. Deshalb: keine Cursortasten benutzen!

Hinweise

Vom IDAS gibt es mehrere verschiedene Versionen, die sich im Detail unterscheiden (Startmeldung, Fehlermeldungen, ...). Die ROM-Versionen enthalten auch verschiedene Zusatzprogramme.

IDAS RAM-Version Kassette R0122

1. Block CRC = DF95, SUM = 0536

2. Block CRC = EF0A, SUM = 51A5
3. Block CRC = 0904, SUM = E84B
4. Block CRC = 6904, SUM = 3FC4

mein Modul idas.rom

1. Block CRC = 7938, SUM = 8F16
2. Block CRC = 1809, SUM = F6E1
3. Block CRC = F7FA, SUM = 47AF
4. Block CRC = B8EF, SUM = C527
5. Block CRC = E250, SUM = D35F

idas.rom = IDAS von Modul M002..M006

Meldung mit „IDAS:„

enthält ZM Z9001 MONITOR V2.0 (ROM) 1985

EPROM (Software für robotron-EPROMMER)

ZM hat Register C und E vertauscht wie in ZM30 der Kassette R0112

IDAS entspricht der Version von Kassette R0122 (nur auf andere Adr. gelinkt)

ROM-Version U. Zander idas_c0.851

1. Block CRC = A02E, SUM = 8BC7
2. Block CRC = 628C, SUM = FBAD
3. Block CRC = A1C7, SUM = 4EA0
4. Block CRC = BBBA, SUM = C4D0
5. Block CRC = B139, SUM = EE30

idas_c0.851 = IDAS von UZ,

Meldung mit „INTERPRETING DIALOG-ASSEMBLER“

enthält ZM Z9001 MONITOR CENT. 1986

QUICK, NORMAL, QLOAD

IDAS entspricht nicht der Version von Kassette R0122

IDAS entspricht nicht der Beschreibung (Seite 3: Nach dem Start meldet sich der Assembler mit der Ausschrift 'IDAS :')

ROM von W. Hasche idas_whasche.rom

1. Block CRC = A02E, SUM = 8BC7
2. Block CRC = 628C, SUM = FBAD
3. Block CRC = A1C7, SUM = 4EA0
4. Block CRC = BBBA, SUM = C4D0
5. Block CRC = 4FDA, SUM = C4F0

die letzten 2K sind vermutlich falsch EPROM5 = EPROM4 (bis auf ein Bit)

damit entspricht idas_whasche.rom höchstwahrscheinlich idas_c0.851

From:

<https://hc-ddr.hucki.net/wiki/> - Homecomputer DDR

Permanent link:

<https://hc-ddr.hucki.net/wiki/doku.php/z9001/software/idas?rev=1387294070>

Last update: **2013/12/17 15:27**

