

Beschreibung

Inhaltsverzeichnis

**INTERPRETATIVER  
DIALOGASSEMBLER**

zum Heimcomputer  
**robotron Z9001**

1.	Einleitung	2
2.	Programmspezifikation	2
3.	Start und Bedienung	3
4.	Arbeitsweise des IDAS	4
4.1.	Eingabe in den Textpuffer	4
4.2.	Sofortausführung von Assemblerbefehlen	5
5.	Beschreibung der Kommandos	5
6.	Einschränkungen und Erweiterungen des Sprachumfanges	12
7.	Fehlermeldungen	12
	<u>Anhang</u>	
	Übersicht zur Assemblersprache SYPS K 1520	A1

VEB ROBOTRON MESSELEKTRONIK >OTTO SCHÖN< DRESDEN  
DDR 8012 Dresden, Lingnerallee 3, Postschließfach 211

digitalisiert von Ulrich Zander im März 2000

5/85 a

Jt 2231/85

## 1. Einleitung

Mit dem Interpretativen Dialogassembler (IDAS) hat der Anwender die Möglichkeit, Programme auf Assemblerniveau zu schreiben. Diese Programme zeichnen sich durch verhältnismäßig geringen Speicherplatzbedarf, hohe Abarbeitungsgeschwindigkeit und volle Ausnutzung der Möglichkeiten des Prozessors aus.

Der IDAS erzeugt einen Maschinencode, der von Rechnern mit dem Schaltkreis U 880 als Prozessor verarbeitet werden kann.

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).

In den folgenden Abschnitten wird die Arbeitsweise des IDAS erläutert. Der Anhang enthält eine Übersicht zur Assemblersprache SYPS K 1520. Ausführliche Angaben zu dieser Sprache sind der entsprechenden Literatur zu entnehmen.

## 2. Programmspezifikation

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

IDAS realisiert folgende Funktionen:

- Erfassen, Einfügen, Ändern und Streichen von Assembleranweisungen in einem internen Textpuffer.
- Übersetzen von Assembleranweisungen in abarbeitsfähige Maschinenbefehle.
- Abarbeiten einzelner Assemblerbefehle sofort nach der Eingabe (interpretativ) oder Abarbeiten des im Textpuffer erfaßten Quellprogramms (ganz oder teilweise).

- Belegen von Speicherplätzen mit gewünschten Werten und Ausgabe von aktuellen Registerinhalten.

## 3. Start und Bedienung

Der Dialogassembler IDAS wird durch die Eingabe seines Namens aufgerufen. Ist das Programm als ROM-Variante gesteckt, so erfolgt der unmittelbare Start, andernfalls muß es von der Kassette eingelesen werden.

Nach dem Start meldet sich der Assembler mit der Ausschrift

**IDAS:**

und fordert die Eingabe eines Pflichtkommandos an:

- Neustart - Eingabe des Zeichens "N".  
Diese Eingabe ist bei Erststart das Programms zur Initialisierung der Zuweisungen des internen Textpuffers und des Maschinencodebereiches zu geben.
- Restart - Eingabe eines beliebigen Zeichens (außer "N"), z. B. auch Betätigen der ENTER-Taste.  
Ein im Textpuffer stehendes Programm bleibt erhalten. Restart ohne vorhergehenden Neustart führt zu undefinierbaren Fehlern.

Durch Ausgabe des Zeichens "+" auf dem Bildschirm wird vom Programm die nächste Bedienereingabe (Assembleranweisung oder Kommando) angefordert.

Jede Tastatureingabe wird erst nach Abschluß der Eingabe mit der ENTER-Taste analysiert und zur Ausführung gebracht. Korrekturen während der Tastatureingabe sind nach Betätigen der Kursortaste <--- möglich.

Erfolgt nach Betätigen von <--- keine weitere Zeicheneingabe, wird die Zeile nicht in den Textpuffer übernommen bzw. die Anweisung nicht ausgeführt.

Die maximal zulässige Zeilenlänge beträgt 64 Zeichen.

Die Adressen des Textpuffers und des Maschinencodebereiches werden bei Programmneustart standardmäßig initialisiert und können mit dem ASGN-Kommando angezeigt und modifiziert werden

#### **4. Arbeitsweise des IDAS**

IDAS ermöglicht zwei verschiedene Arbeitsweisen. Zum einen können Assembleranweisungen (Befehle und Definitionsanweisungen) über Tastatur oder vom Magnetband in einen internen Textpuffer eingegeben, aufbereitet und anschließend übersetzt und abgearbeitet werden. Zum anderen können über Tastatur eingegebene Assembleranweisungen (Befehle) sofort nach der Eingabe übersetzt und ausgeführt werden.

##### **4.1. Eingabe in den Textpuffer**

Die Abspeicherung der Assembleranweisungen im Textpuffer erfolgt in einer speziellen Form zusammen mit einer Zeilennummer. Der Textpuffer ist nach aufsteigenden Zeilennummern organisiert. Bei Eingabe über die Tastatur werden Anweisungen mit einer vorangestellten Zeilennummer in den Textpuffer übernommen und an entsprechender Stelle eingeordnet.

Beispiel:

```
+ 110 LD A,5
```

Eine lückenhafte Numerierung der Anweisungen ermöglicht das einfache Einfügen. Das Ändern ist durch eine Eingabe mit gleicher Zeilennummer möglich.

Das Streichen einzelner Anweisungen erfordert lediglich die Eingabe der entsprechenden Zeilennummer.

Jede eingegebene Anweisung wird vor ihrer Abspeicherung einer Syntaxprüfung unterzogen (keine Operandenberechnung).

Nach Eingabe eines RUN- oder ADDR-Kommandos oder einer Anweisung ohne Zeilennummer wird das im Textpuffer gespeicherte Quellprogramm geschlossen übersetzt. Der Maschinencode (Befehle, Konstanten und Bereiche) wird im Maschinencodebereich aufgebaut. Durch das RUN-Kommando wird dieser anschließend ab der ersten spezifizierten Anweisung gestartet und ganz oder teilweise abgearbeitet. Nach der

letzten spezifizierten Anweisung generiert IDAS einen Rücksprung zu sich selbst.

##### **4.2. Sofortausführung von Assemblerbefehlen**

Assemblerbefehle, deren Eingabe über Tastatur ohne Zeilennummer erfolgt, werden sofort übersetzt und ausgeführt (interpretative Arbeitsweise).

Beispiel: + LD A,5

Dabei kann auf im Textpuffer definierte Konstanten, Bereiche und Unterprogramme Bezug genommen werden.

Die Befehle DI und

LD I,A dürfen in dieser Arbeitsweise nicht verwendet werden, da die Tastaturbedienung des Betriebssystems interruptgesteuert ist und durch diese Befehle außer Funktion gesetzt wird.

#### **5. Beschreibung der Kommandos**

ADDR Anzeige eines Adreßwertes

Syntax: ADDR ausdruck

Das Kommando zeigt den Wert des angegebenen Ausdrucks in hexadezimaler Form an. Für Symbole wird ihr aktueller Adreßwert eingesetzt. Dazu wird erforderlichenfalls der Textpufferinhalt übersetzt.

Beispiel:

```
ADDR SYMB  
ADDR 1000H-ANZ-8
```

## ASGN Anzeigen und Modifizieren der Pufferadressen

Syntax: ASGN

Das Kommando zeigt die aktuellen Adressen des Textpuffers und des Maschinencodebereiches an:

```
TB(XXXXH-XXXXH,XXXXH):  
MC(XXXXH-XXXXH,XXXXH):
```

Die Adressen werden in der Reihenfolge Anfangsadresse aktueller Füllstand - Endadresse angezeigt. Die Anzeigen sind durch [ENTER] zu quittieren (keine Änderung) oder durch die Eingabe neuer Anfangs- und Endadressen zu ändern.

Beispiel:

```
+ ASGN  
TB (2000H-2000H,7FFFH) :  
MC (0400H-0400H,1FFFH) : 500H, FFFH  
+
```

Nach einer Textpufferzuweisung ist das vorhergehende Quellprogramm nicht mehr verfügbar. Der Maschinencodebereich-Füllstand zeigt erst nach einer Übersetzung des Textpuffers den aktuellen Wert an.

## CM Monitorkommando

Syntax: CM

Das Kommando bewirkt den Rücksprung in das Monitorprogramm. Die Rückkehr in IDAS ist durch das Monitorkommando IDAS möglich.

## DR Anzeige der Register

Syntax: DR

Das Kommando bewirkt die Anzeige der aktuellen Registerinhalte. Diese werden durch IDAS selbst nicht beeinflusst.

## DS Anzeige und Modifizierung von Speicherinhalten

Syntax: DS

Das Kommando ermöglicht die Anzeige und Änderung des Inhalts von Speicherplätzen. Nach der Eingabeaufforderung ":" ist die Adresse des anzuzeigenden Speicherplatzes einzugeben. Die Anzeige erfolgt in der Form "Adresse Inhalt -". Nach dem Bindestrich kann ein neuer Speicherinhalt eingegeben werden. Durch Betätigen der ENTER-Taste wird die Änderung übernommen bzw., falls keine Eingabe erfolgte, der alte Speicherinhalt beibehalten.

Durch Betätigen von ↓ wird der Inhalt der Speicherzelle mit der nächsthöheren Adresse angezeigt, bei Betätigen von ↑ der Inhalt der Speicherzelle mit der nächstniedrigeren Adresse. Die Abarbeitung des Kommandos wird mit der STOP-Taste beendet.

Beispiel:

```
+ DS  
: 400H  
4,00H 3EH- [ENTER]  
401H 03H-04H [ENTER]  
402H 06H- ↑  
401H- 04H- ↑  
400H 3EH- ↓  
401H 04H- [STOP]
```

## KILL Streichen von Zeilen im Textpuffer

Syntax: KILL m,n - von Zeile m bis Zeile n  
KILL n - Zeile n  
KILL - alle Zeilen

Das Kommando bewirkt das Streichen von Zeilen im Textpuffer. Die Zeilen werden durch ihre Zeilennummern spezifiziert. Bei lückenhafter Zeilennumerierung wird das Kommando KILL m,n interpretiert als "Streichen von Zeile m bzw. nächstfolgender bis Zeile n bzw. vorhergehender".

## LIST Anzeige des Textpufferinhalts

Syntax: LIST m,n        - von Zeile m bis Zeile n  
         LIST n           - Zeile n  
         LIST            - alle Zeilen  
         LIST marke      - alle Zeilen ab Marke

Das Kommando bewirkt die formatierte Ausgabe von Zeilen des Textpuffers auf dem Bildschirm. Das Listen kann mit [CONTR] [S] unterbrochen, mit einer beliebigen Taste fortgesetzt und mit [STOP] abgebrochen werden.

## MCSV Schreiben von Maschinencode auf Magnetband

Syntax: MCSV [anfangsadresse,endadresse]  
         MCSV [marke 1,marke 2]

Durch dieses Kommando wird der im MC-Bereich befindliche Maschinencode auf Magnetband abgespeichert. Werden Anfangs- und Endadresse nicht eingegeben, verwendet IDAS die Anfangs- und Endadresse des MC-Bereiches. Statt Anfangs- und Endadresse können ebenfalls Marken aus dem Quelltext angegeben werden, die diesen Adressen entsprechen.

Nach der Eingabeanforderung "FILENAME: " kann ein 1 bis 8 Zeichen langer Name vergeben werden, unter dem der Maschinencode, auf Magnetband abgespeichert wird.

Die Startadresse des Programme kann nach der Eingabeanforderung "STARTADRESSE: " eingegeben werden. Wird die Abfrage nur mit [ENTER] quittiert, setzt IDAS die Anfangsadresse als Startadresse.

Die abgespeicherten Dateien sind immer vom Typ ".COM", d. h., sie werden beim Einlesen mit dem Monitorprogramm stets bei ihrer Startadresse gestartet. Sollen Datendateien ausgelagert werden, muß folglich als Startadresse 0FFFFH angegeben werden.

Vor Adressen, deren erstes Zeichen ein Buchstabe ist, sind führende Nullen einzugeben.

Im Anschluß an das Abspeichern kann nach der Ausschrift "VERIFY (Y/N)? " zur Kontrolle ein vergleichendes Lesen ausgeführt werden.

### Beispiel:

```
+ MCSU 500H,7FFH  
FILENAME: TEST1  
STARTADR.: 0FFFFH  
  
start tape
```

## READ Eingabe in den Textpuffer

Syntax: READ [zeilennummer]  
         READ [marke]  
         READ

Das Kommando ermöglicht die Eingabe von Assemblerzeilen oder ganzen Programmen. Die eingegebenen Zeilen werden nach der spezifizierten Marke oder Zeilennummer in das vorhandene Programm eingefügt. Wird keine Zeilennummer oder Marke angegeben, werden neue Zeilen vor der ersten schon vorhandenen Zeile eingeordnet.

Nach der Eingabeanforderung "DEVICE (T/(C))" ist das Eingabegerät festzulegen. Dabei kann eingegeben werden:

C - Tastatur (Standardgerät)  
T - Magnetband .

Bei Eingabe über die Tastatur wird die Zeilennummer intern bereitgestellt, d. h., sie ist nicht einzugeben. Vom Magnetband können Dateien vom Typ ".TXT" eingelesen werden.

Beispiel:

```
+      READ UP1
DEVICE (T/(C))? T
FILENAME: DAT1

start tape
```

Die auf Magnetband befindliche Datei DAT1 muß positioniert und das Einlegen mit [ENTER] gestartet werden. Die eingelesenen Assemblerzeilen werden nach der Zeile mit der Marke UP1 in den Textpuffer eingefügt.

RUN Abarbeiten von Anweisungen aus dem Textpuffer

Syntax: RUN m,n           - von Zeile m bis Zeile n  
          RUN n            - Zeile n  
          RUN              - alle Zeilen

Das im Textpuffer gespeicherte Quellprogramm wird vollständig übersetzt. Anschließend wird der erzeugte Maschinencode an der zur ersten spezifizierten Anweisung gehörenden Adresse gestartet und abgearbeitet. Nach dem zur letzten spezifizierten Anweisung gehörenden Maschinencode generiert IDAS einen Rücksprung zu sich selbst, so daß sich nach der Abarbeitung der Anweisungsfolge im Normalfall wieder IDAS mit einer neuen Eingabeaufforderung "+" meldet.

Deshalb ist darauf zu achten,

- daß bei der Abarbeitung des Maschinencodes vom Start bis zum IDAS-Rücksprung nur Befehle abgearbeitet werden (keine Konstanten oder Bereiche),
- daß die Befehlsfolge nicht vor ihrem physischen Ende (=IDAS-Rücksprung) verlassen wird.

STEP Neunummerierung der Anweisungen im Textpuffer

Syntax: STEP - Schrittweite 1  
          STEP n - Schrittweite n ( $1 \leq n < 256$ )

Alle Im Textpuffer befindlichen Anweisungen werden mit der angegebenen Schrittweite fortlaufend neunummeriert.

WRIT Ausgabe aus dem Textpuffer

Syntax: WRIT [zeilennummer]  
          WRIT [marke]

Das im Textpuffer befindliche Assemblerprogramm wird ab der angegebenen Marke oder Zeilennummer auf das Ausgabegerät ausgegeben. Sind Marke oder Zeilennummer nicht spezifiziert, erfolgt die Ausgabe des gesamten Textpufferinhalts.

Nach der Kommandoeingabe ist entsprechend der Eingabeaufforderung "DEVICE (T/(C))?"

- T für Magnetband oder
- C für Bildschirm (Standardgerät) als Ausgabegerät einzugeben.

Bei Ausgabe auf Magnetband ist nach der Eingabeaufforderung "FILENAME:" der Name (1 bis 8 Zeichen) anzugeben, unter dem die Datei abgespeichert werden soll. Der Typ der Datei ist ".TXT".

Die Ausgabe auf dem Bildschirm entspricht der des LIST- Kommandos, jedoch ohne Angabe der Zeilennummern. Im Anschluß an das Abspeichern kann nach der Ausschrift "VERIFY (Y/N)?" zur Kontrolle ein vergleichendes Lesen ausgeführt werden.

Beispiel:

```
+      WRIT MA1
DEVICE (T/(C))? T
FILENAME: TEST

start tape
```

## 6. Einschränkungen und Erweiterungen des Sprachumfanges

IDAS realisiert den gesamten Befehlsumfang des K 1520 in der Assemblersprache SYPS K 1520.

Nicht zugelassen sind:

- die Assembleranweisungen DEF, EJEC, TITL, IF, ENIF, MACR, ENDM,
- externe Symbole,
- Oktalzahlen, von mehr als 255 Bytes,
- zusammengesetzte Ausdrücke in Bytespezifikationen, z. B. (L(ADR+8),
- Kommentare (werden ignoriert).

Beim Gebrauch von Assembleranweisungen ist weiterhin zu beachten:

- Symbole auf der rechten Seite von EQU-, ORG- und BER-Anweisungen müssen vor ihrem Auftreten definiert sein, andernfalls erhalten sie den Wert 0 zugewiesen.
- Bei ORG-Anweisungen wird vor dem Setzen des Speicherplatzzuweisungszählers kontrolliert, ob die definierte Adresse innerhalb des Maschinencodebereiches liegt.
- Die Anweisungen PN und END können mit in den Textpuffer eingegeben werden, sie haben jedoch für den Ablauf des Übersetzungsvorgangs im IDAS keine Bedeutung.

## 7. Fehlermeldungen

- +OPD+ (Operand ERROR)  
Operandenfehler; Fehler im Operandenteil des Befehls
- +MNE+ (Mnemonic ERROR) Mnemonikfehler; unbekannter Assembler oder Kommandocode
- +MDS+ (MULTIPLE DEFINED Symbol)  
Mehrfach definiertes Symbol
- +NDS+ (NOT DEFINED Symbol)  
Nicht definiertes Symbol
- +SYM+ (Symbol ERROR) Fehlerhaftes Symbol; das im Namensteil der Anweisung definierte Symbol ist syntaktisch fehlerhaft.
- +LNR+ (LINENUMBER ERROR)

Unzulässige Zeilennummer; die angegebene Zeilennummer ist syntaktisch fehlerhaft oder nicht im Textpuffer vorhanden.

- +LNG+ (LENGTH ERROR)  
Längenfehler; die über die Tastatur eingegebene Zeichenkette ist länger als 64 Zeichen.
- +ILO+ (ILLEGAL OPEPATION)  
Unzulässige Operation; bei der interpretativen Abarbeitung wurde eine nicht sinnvolle Anweisung gegeben (Pseudoanweisungen, Relativsprünge).
- +TB++ (TEXTPUFFER)  
Bei der Programmeingabe wurde das Ende des Textpuffers erreicht.
- +MC++ (MASCHINE CODE)  
Bei der Generierung des Maschinencodes wurde das Ende des Maschinencodebereiches erreicht. Dieser Fehler erscheint auch, wenn durch eine ORG-Anweisung im Programm Speicherplatz außerhalb des Maschinencodebereiches belegt werden soll.

Weiterhin können Fehler beim Einlesen von Kassette auftreten. Die zur Beseitigung erforderlichen Maßnahmen sind dem Programmierhandbuch des Heimcomputers robotron Z 9001 zu entnehmen.