

```

*****
*                                     ZUSATZ-MONITOR                               *
*                                     für Computer robotron Z9001                       *
*****

```

## 0. Einleitung

Mit dem vorliegenden ZUSATZ-MONITOR (Debugger) für den Computer "robotron Z9001" wird besonders dem Nutzer ein leistungsfähiges Hilfsmittel geboten, das Programme auf der Ebene der Maschinensprache auf dem Computer implementieren will. Dazu stehen bereits leistungsfähige Entwicklungshilfen, z.B.: IDAS, EDITOR/ASSEMBLER (EDAS), zur Verfügung. Das Testen von Programmen, die mit den genannten Entwicklungshilfen erarbeitet wurden, wird durch den (ZUSATZ-) MONITOR wesentlich erleichtert.

Der MONITOR bietet aber auch die Möglichkeit, direkt Maschinencode in den Speicher des Rechners einzugeben und ablaufen zu lassen.

Insgesamt stehen 20 Kommandos zur Verfügung, mit denen die wesentlichen Manipulationen beim Installieren und Testen neuer Programme ausgeführt werden können.

Die effektive Arbeit mit dem MONITOR setzt gute Kenntnisse der Rechnerstruktur sowie der Befehlsstruktur des Prozessors U880 voraus.

Zusammen mit den o.g. Entwicklungshilfen kann der MONITOR aber auch ein wertvolles Hilfsmittel für das Einarbeiten in das Gebiet der Maschinenprogrammierung werden und zum besseren Verständnis der Arbeitsweise des Prozessors U880 beitragen.

Der MONITOR steht als Maschinencodeprogramm auf Kassette in drei Varianten entsprechend der verfügbaren RAM-Konfiguration und als ROM-Version bereit.

Speicherbelegung:		3000H	bis 3FFFH	- ZM30
	oder	7000H	bis 7FFFH	- ZM70
	oder	A800H	bis B7FFFH	- ZMA8
		220H	bis 22FH	- (EXEC-Sprungvektoren)

Kommandos: 20

Zusätzlich bedienbare	- Lochbandleser	(SIF-1000)
Peripherie	- Lochbandstanzer	(SIF-1000)
(über PIO1B, siehe	- SIF-1000-Drucker	
hierzu Abschnitt 2):	- V-24-Drucker	

Laden/Start des Monitors:

IM OS-Modus des "robotron Z9001" (nach Einschalten, RESET, 'BYE' im BASIC oder 'CM' im IDAS) wird eingegeben:

**ZMxx** <ENTER>                    xx entsprechend der RAM-Konfiguration

Der MONITOR meldet sich nach dem Laden, dem Löschen des Bildschirms und der Ausgabe einer Überschrift mit dem PROMPT-Zeichen => und ist damit bereit zur Kommandoeingabe.

Der MONITOR kann im OS-Modus des "robotron Z9001" mit

**ZM** <ENTER>

erneut gestartet werden.



- C (Convert):** Umrechnung von Dezimalzahlen in Hexadezimalzahlen und umgekehrt.  
 CD<dezimalzahl>{\_^}            Dez. --> Hex.  
 CH<hexzahl>{\_^}                Hex. --> Dez.  
 CB<binärzahl>{\_^}            Bin. --> Hex.  
 Beispiel: CD100\_
- D (Display):** Ausgabe eines Speicherbereiches auf Konsole.  
 D<anfadr>\_[<endadr>]{\_^}  
 Beispiel: D1000\_10FF\_
- E (Endeblock):** Ausgabe eines Endeblockes auf das Punch-Gerät.  
 E[adresse]{\_^} (nicht sinnvoll bei AP=T)
- F (Fill):** Beschreiben eines Speicherbereiches mit einem konstanten Wert.  
 F<anfadr>\_<endadr>\_wert{\_^}
- G (Go):** Starten eines Anwenderprogrammes mit max. zwei Haltepunkten.  
 G<startadr>[\_<break1>][\_<break2>]{\_^}  
 Beispiel: GC00D
- H (Hex.-Math.):** Bildung der Summe und der Differenz zweier Hexadezimalzahlen.  
 H<hexzahl1>\_<hexzahl2>{\_^}  
 (Ausnahme: H0\_0)
- J :** Nichtzerstörender Speicherkurztest.  
 J<anfadr>\_<endadr>{\_^}
- M (Move):** Transport eines Speicherbereiches auf einen anderen Speicherbereich.  
 M<anfadr>\_<endadr>\_<zieladr>{\_^}
- N (Null):** Ausgabe von binären Nullen auf den Punch-Kanal.  
 (Lochbandvorschub, sinnlos bei AP=T)
- P (Punch):** Ausgabe eines Speicherbereiches im INTEL-Hex-Format auf den Punch-Kanal.  
 P<anfadr>\_<endadr>\_{\_^} (<startadr> nur bei AP=T)
- Q (Query):** Lesen und Schreiben auf IO-Ports.  
 QI<portadr>{\_^}                    Lesen vom Port  
 QO<portadr>\_<byte>{\_^}            Schreiben zum Port  
 Beispiel: QI89H\_  
           QO88\_4
- R (Read):** Einlesen eines INTEL-Hex-Files über den Reader-Kanal.  
 R[<base>]{\_^}
- S (Substitute):** Modifizieren von Speicherzellen. Mit der Leertaste kann zur Adresse weitergeschaltet werden. Mit <ENTER> wird das Kommando beendet.  
 S<anfadr>{\_^}
- T (Type):** Ausgabe eines Speicherbereiches in ASCII-Darstellung (falls möglich, sonst .).  
 T<anfadr>\_[<endadr>]{\_^}
- V (Verify):** Vergleich eines Speicherbereiches mit einem anderen.  
 V<anfadr>\_<endadr>\_<zieladr>{\_^}
- W (Write):** Schreiben von ASCII-Zeichen in den Speicher. Mit <-- (Cursor links) kann zurückgeschritten werden. Mit <STOP> wird die Eingabe beendet. Die Adresse des zuletzt eingegebenen Zeichens +1 wird vom Monitor ausgegeben.  
 W<anfadr>{\_^}

**X :** Ausgabe des Z80-Registersatzes  
 X^ Ausgabe des 1. Registersatzes  
 X'^ Ausgabe des 2. Registersatzes  
 X['<reg> Ausgabe und Änderungsmöglichkeit für Register <reg>

**Y :** Suchen nach einer Zeichenfolge von max. 255 Bytes  
 Y<byte>[\_<byte>][...]^  
 Beispiel: Y42,41,53,49,43^

**Z :** Ausgabe der höchsten RAM-Adresse (RAM-TOP)

Für die Kommandos I, K, L, O und U stehen ab 220H fünf Sprungvektoren, über die durch Modifikation der Adressen zusätzliche Kommandoroutinen angebonden werden können.

Fortlaufende Bildschirmausgaben können durch Betätigen der Leertaste gestoppt und mit erneutem Betätigen fortgesetzt werden. Die STOP-Taste führt zum Abbruch des Kommandos.

Wird bei der Parametereingabe zu einem Kommando ein Fehler erkannt, bevor \_ oder ^ eingegeben wurde, so kann eine Korrektur der Eingabe durch Weiterschreiben erreicht werden, da nur die letzten vier Zeichen zu jedem Parameter gewertet werden.

Beispiel: D100F1000\_20FF10FF  
 entspricht D1000\_10FF

## 2. Anschluß von peripheren Geräten

Der freie PIO-Port (PIO1/B), der an einer 15poligen EFS-Buchse an der rechten Seite des Computers gemeinsam mit zwei Signalen des ebenfalls noch für den Anwender nutzbaren CTC-Kanals (CTC1) zur Verfügung steht, kann sehr einfach für den Anschluß zusätzlicher peripherer Geräte, wie Lochbandleser bzw. -stanzer und Drucker verwendet werden.

Der MONITOR bietet hierfür bereits entsprechende Software-Routinen an, die das Betreiben von Geräten mit SIF-1000-Interface ermöglichen.

### 2.1. Belegung der PIO-Buchse für SIF-1000-Anschluß

	<b>A</b>	<b>B</b>	<b>C</b>
1	0	0	/END
2	/DAT1	/DAT2	/DAT3
3	/DAT4	/DAT5	/DAT6
4	/DAT7	/RUF	
5	+5V		

/RUF ist stets ein abgehende, /END ein ankommendes Signal. Beim SIF-A-Kanal sind die Datenleitungen /DAT1 abgehende, sonst ankommende Signale.

Zu beachten sind unbedingt Signalpegel und Lastbedingungen!

Beim 12-V-SIF-1000 müssen die ankommenden Signale auf  $\leq 5V$  begrenzt werden (Dioden gegen +5V).

### 2.2. Geräte-Treiber-Routinen

Der MONITOR stellt folgende Geräuteroutinen zur Verfügung:

<b>Routine</b>	<b>Assign-Kommando</b>	<b>zugehörige Adresse in OS-I/O-Tabelle</b>
SIF-A	AP=P	0EFDB
	AL=U	0EFE7
SIF-E	AR=P	0EFD3
V24-A	AP=U	0EFD3
	AL=L	0EFE5

Bei den SIF-1000-Routinen erfolgt die Übertragung mit 7 Datenbits ohne Parität. Das 8. Datenbit des PIO-Ports wird für die Bildung des RUF-Signals verwendet. Damit kann in den meisten Fällen auf jegliche zusätzliche Hardware für den Geräteanschluß verzichtet werden.

Sie Flanken des END-Signals werden mit dem Kanal 1 des CTCs erfaßt (Betrieb als Zähler). Die SIF-E-Routine wertet nur die HL-Flanke des END-Signales aus und ist besonders für die Bedienung eines Bandlesers geeignet.

Die V24-Ausgabe-Routine ist für das Betreiben eines Thermodruckers "robotron K6303" (TD40) vorgesehen.

Die Übertragungsgeschwindigkeit beträgt 1200 Baud. Es werden 7 Datenbits, ein START- und ein STOP-Bit übertragen.

Der Drucker muß für die Betriebsart XON/XOFF eingestellt werden.

## ANHANG

### Kurzübersicht über die MONITOR-Kommandos

<b>A</b> (ssign)	A<logisches Gerät>=<physisches Gerät> (C, R, P, L)      (C, V, B, P, T, L, U)
<b>B</b> (ye)	
<b>C</b> (onvert)	CD<dezimalzahl>[_^]      Dez. --> Hex. CH<hexzahl>[_^]      Hex. --> Dez. CB<binär>[_^]      Bin. --> Hex.
<b>D</b> (isplay)	D<anfadr>[_<endadr>][_^]
<b>E</b> (of)	E[adresse][_^]      (nicht sinnvoll bei AP=T)
<b>F</b> (ill)	F<anfadr>[_<endadr>]_wert[_^]
<b>G</b> (o)	G<startadr>[_<break1>][_break2] [_^]
<b>H</b> (ex)	H<hexzahl1>[_<hexzahl2>][_^]
<b>J</b>	J<anfadr>[_<endadr>][_^]
<b>M</b> (ove)	M<anfadr>[_<endadr>]_<zieladr>[_^]
<b>N</b> (ull)	
<b>P</b> (unch)	P<anfadr>[_<endadr>][_^] (<startadr>)
<b>Q</b> (uery)	QI<portadr>[_^]      Lesen vom Port QO<portadr>[_<byte>][_^]      Schreiben zum Port
<b>R</b> (ead)	R[<base>][_^]
<b>S</b> (ubstitute)	S<anfadr>[_^]
<b>T</b> (ype)	T<anfadr>[_<endadr>][_^]
<b>V</b> (erify)	V<anfadr>[_<endadr>]_<zieladr>[_^]
<b>W</b> (rite)	W<anfadr>[_^]
<b>X</b> (eXamine)	X^      Ausgabe des 1. Registersatzes X'^      Ausgabe des 2. Registersatzes
<b>Y</b>	Y<byte>[_<byte>][...]^      (suchen)
<b>Z</b>	(RAM-TOP)

#### Aufbau des Flagregisters:

```
*****
* Bit            7            6            5            4            3            2            1            0            *
*-----*
*               Sign       Zero       -       H       -       P/U       N       Carry *
*****
```