

Anzeigezuordnung

```

-----
Anzeige der      Adressen oder      Daten
Programmebene   Systeminformationen
---            ---            ---
| |      | |      |E|      |I|      |n|      |G|      |=|      | |
---            ---            ---            ---            ---
                                0      1      2      3      4      5
                                Cursorpositionen
    
```

```

=====
Bedienung der Software
=====
    
```

Anfangszustand:

```

0 0 0 0 0 0 0 1   Kontroll-LED 0 leuchtet

- - - - - - - -   o keine Programmanzeige in der
      E i n g :   Systeminformation
- - - - - - - -   o Aufforderung zur Eingabe der
                  Unterprogrammebene
    
```

Bsp: Eingabe von 0

```

0 0 0 0 0 0 1 1   Anzeige der Ebene 1

- - - - - - - -   o Programm - Debugger
D B   E i n g :   o Aufforderung zur Eingabe des
- - - - - - - -   gewuenschten Programms
    
```

Bsp: Eingabe von 1

```

0 0 0 0 0 0 1 1   Anzeige der Ebene 2

- - - - - - - -   o Programm - Debugger - Datenanzeige
D A   1 2 0 0   3.7 o auf Cursorposition 4 koennen
- - - - - - - -   Daten geaendert werden
    
```

moegliche Programmschleifen nach dem Einschalten



- 0. DB - Debugger
Programmentwicklung und Programmtest auf der Hardware dieses MK
- 1. TB - Tonband
Lesen und Schreiben von Programmen mit dem Tonband
- 2. EP - EPROM Programmiergeraet
EPROM Programmiergeraet fuer die Typen 2716 , 2732
- 3. LA - Logikanalysator
Schritt- und Laufstest beliebiger anderer MK auf deren Hardware (mit WAIT-Zyklen)

4. HT - Hardwaretest

kompletter Hardwaretest anderer MK ueber ihr Bussystem
mit Beschreiben des RAM um die Software auf anderen
MK Testen zu koennen

0. Debugger

Es werden vom Debugger folgende Moeglichkeiten geboten:

0.0. - Adresse setzen

Systemanzeige: AD Kursoranfang: 0 Kursorende: 3
Im Adressfeld kann eine neue Adresse eingegeben werden.
Standard: 1200H
+ = Cursor nach rechts
- = Cursor nach links
BA = zurueck zur Unterebene 0.

0.1. - Daten auf der Adresse anzeigen

Systemanzeige: DA Kursoranfang: 4 Kursorende: 5
Es werden die Daten auf der Adresse (0.0.) angezeigt.
+ = Cursor nach rechts.Ist der Cursor auf Position
6 so erfolgt ein Inkrementieren der Adresse.Der Cursor
steht danach auf Position 5.
- = Cursor nach links.Ist der Cursor auf Position
5 so erfolgt ein Dekrementieren der Adresse. Der
Cursor steht danach auf Position 5.
ET= Inkrementieren der Adresse Der Cursor steht danach auf
Position 5.
BA= zurueck zur Unterebene 0.

0.2. - CPU-Hilfsregister loeschen

Systemanzeige: RL
Es werden die internen CPU-Hilfsregister geloescht.
(danach Standardeinstellung wie bei Kaltstart)

0.3. - Register anzeigen

Systemanzeige: RA Kursoranfang: 0 Kursorende: 3
Es erfolgt ein Weiterschalten auf das Hilfsmenu
(Ebene3)
0.3.0. - Register A F
0.3.1. - Register B C
0.3.2. - Register D E
0.3.3. - Register H L
0.3.4. - Register A'F'
0.3.5. - Register B'C'
0.3.6. - Register D'E'
0.3.7. - Register H'L'
0.3.8. - Register I X
0.3.9. - Register I Y
0.3.A. - Register S P
0.3.B. - Register I R
0.3.C. - Speicher (HL)

+ = Cursor nach rechts
- = Cursor nach links
BA = zurueck zur Hilfsebene 0.3.

0.4. Breakpoint setzen

MLBA.TXT

Systemanzeige: BS Kursoranfang: 0 Kursorende: 3
Bedienung wie 0.0. nur mit Eintragen des
Restartbefehls (RST 6) auf die Breakpointadresse

0.5. Schrittbetrieb

Systemanzeige: ST Kursoranfang: 0 Kursorende: 0
Mit dieser Betriebsart koennen beliebige Programme im
Schrittest betrieben werden. Der Start erfolgt ab Adresse
nach 0.0. Auf dem Adressfeld kann die Adresse und auf dem
Datenfeld das Datenwort des naechsten abzuarbeitenden
Befehls abgelesen werden.
Innerhalb dieser Betriebsart kann mit ET die Abarbeitung
des naechsten Befehls erfolgen.

ET - weiter im Schrittbetrieb
BA - zurueck zum Debugger

0.6. Laufbetrieb

Systemanzeige: LT Kursoranfang: 0 Kursorende: 0
Aufruf des Programmes ab Adresse nach 0.0. Ende des Pro-
gramms durch Erreichen der Breakpointadresse oder nach
RET durch Auffangen durch den Debugger. Anschliessend
erfolgt der Einsprung in die Beriebsart Daten anzeigen.
(0.1.)

1. Tonband

1.0. Dateinummer

Systemanzeige: DN Kursoranfang: 0 Kursorende: 3
Eingeben der Dateinummer der zu lesenden oder zu
schreibenden Datei

1.1. Anfangsadresse

Systemanzeige: AA Kursoranfang: 0 Kursorende: 3
Eingeben der Anfangsadresse ab der die Datei gelesen
oder geschrieben werden soll

1.2. Endadresse

Systemanzeige: EA Kursoranfang: 0 Kursorende: 3
Eingeben der Endadresse ab der die Datei gelesen
oder geschrieben werden soll

1.3. Blockendezahl

Systemanzeige: BE Kursoranfang: 0 Kursorende: 3
Eingeben der Anzahl der zu lesenden oder zu schreibenden
Bloেকে

1.4. Blockanfangszahl

Systemanzeige: BA Kursoranfang: 0 Kursorende: 3
Eingeben der ersten Blocknummer der zu lesenden oder zu
schreibenden Datei

1.5. Berechnung der Blockendezahl

Aus der Anfangsadresse und der Endadresse wird die Block-
endezahl berechnet

1.6. Datei schreiben

Nach Ablauf des Einschwingtones (etwa 2 s) erfolgt die
Ausgabe der Datei auf Tonband. Auf der Cursorposition 5
wird dabei die aktuelle Blocknummer angezeigt.

1.7. Datei lesen

Der Lesevorgang muss innerhalb des Einschwingtones gestartet werden. Sind Dateinummer und Blocknummer richtig erkannt worden, so erfolgt das Einlesen der Datei und auf der Cursorposition 5 die Anzeige der gelesenen Blocknummer. Stimmen die gelesene Dateinummer oder die gelesene Blocknummer nicht mit den Vorgaben ueberein (oder wurde ein Lesefehler erkannt) so wird auf den Cursorpositionen 0-3 die gelesene Blocknummer und auf den Cursorpositionen 4-5 die gelesene Dateinummer angezeigt.

2. EPROM-Programmierung

2.0. Anfangsadresse im RAM

Systemanzeige: AR Kursoranfang: 0 Kursorende: 3
Eingeben der Anfangsadresse des Speicherbereiches im RAM mit dem gearbeitet werden soll.

2.1. Endadresse im RAM

Systemanzeige: ER Kursoranfang: 0 Kursorende: 3
Eingeben der Endadresse des Speicherbereiches im RAM mit dem gearbeitet werden soll.

2.2. Anfangsadresse im EPROM

Systemanzeige: AE Kursoranfang: 0 Kursorende: 3
Eingeben der Anfangsadresse auf dem EPROM ab der gearbeitet werden soll. (0000H ist Anfang)

2.3. Test

Systemanzeige: ET
Beim gesteckten EPROM wird getestet ob alle Adressen die Datenbelegung 0FFH aufweisen (EPROM geloescht)
-wenn ja so erfolgt die Aufschrift "EP.I.O."
-wenn nicht so erfolgt die Aufschrift "LOES"

2.4. Lesen

Der EPROM wird ab Adresse "AE" in den RAM ab Adresse "AR" bis Adresse "ER" gelesen.

2.5. Programmieren

Der RAM Inhalt wird von Adresse "AR" bis Adresse "ER" auf den EPROM ab Adresse "AE" gebrannt.