

```

*****
**
**          EPROM- Programmierung  mc80.30          **
**
**          Version 3.0.                          **
**

```

```

*****
** 04.11.1985 ** Bearbeiter: Andreas Schimpf / Gera 6224190 **
*****

```

\*\*\*\*\* VEREINBARUNGSTEIL \*\*\*\*\*

```

EPRX EQU 000E0H
OUTX EQU 00BBEH ;UP: Text auf Display schreiben
MBMX EQU 00BC1H ;UP: Text auf Display schreiben
TABX EQU 00BCDH ;UP: Tastatur abfragen
ZLEX EQU 00BD0H ;UP: Zahl von Textpuffer lesen
BFEX EQU 00BD6H ;UP: Bildfenster festlegen
LIYX EQU 00BE8H ;UP: IY- Vektor laden
HBSX EQU 00BEEH ;UP: Byte auf Textpuffer schr.
CONX EQU 00BEEH ;UP: Consolentreiber
LZVTX EQU 0FB62H
ST08X EQU 0CB70H ;Steuerwort fuer 2708
ST16X EQU 0CA32H ;Steuerwort fuer 2716
ST32X EQU 0CD6CH ;Steuerwort fuer 2732

```

\*\*\*\*\* HAUPTPROGRAMM \*\*\*\*\*

```

PARA: LD HL,TPUF ;Arbeitsbereich initialisieren
      PUSH IY
      PUSH IX
PI00: LD B,018H.
      LD M,000H
      INC HL
      DJNZ PI00
      LD A,(DE)
      CMP OFFH ;z.Z. nur mit Parameterabfrage
      JPNZ PART ;Parameter abf.
      LD (KOPU),DE ;Kommandopufferanfang
      PUSH DE
      EX DE,HL
      LD M,020H
      INC HL
      LD (ADPU),HL ;aktuelle Position
      JR PARA
TST1: LD HL,FLAG ;Behandlung im Korrekturmodus
      BIT 0,M
      SET 0,M
      SET 7,M
      PUSH AF
      LD HL,(KOPU) ;aktuelle Position ermitteln
      INC HL
      LD (ADPU),HL
      PUSH HL
      PUSH DE
      LD DE,(ETYP)
      CALL HDS ;EPROM- Typ auf Display schr.
      POP DE
      POP HL
      POP AF

```

*AZ löschen*

*z.Z. nur mit Parameterabfrage  
Parameter abf.*

```

JRZ      PAR0
LD       HL, (KOPU)      ;fertig
LD       M, OFFH
POP      DE
PART:    POP      IX
POP      IY
OR       A
RET

PAR0:    LD       HL, TXTA      ;Text: 'EPROM- Typ angeben ...'
CALL     TAUS             ;ausschreiben
LD       DE, 01807H      ;Displayposition
CALL     Z4DE             ;Zahl 4- stellig, hexadezimal,
                          ;eingeben
;
JRC      TST1             ;Fehler
LD       A, H
CMP      027H
LD       IX, PAR0        ;Sprungadresse bei Fehler
JRNZ    PARF
LD       A, L
CMP      008H             ;2708 ?
JRZ     PAR1
CMP      016H             ;2716 ?
JRZ     PAR1
CMP      032H             ;2732 ?
JRZ     PAR1
PARF:    LD       DE, 01710H    ;Position fuer Fehlerausschrift
LD       HL, FLAG
SET     7, M
LD       HL, TXFE        ;Text: 'Fehler !'
LD       BC, 00008H
CALL    OUT              ;ausschreiben
JMP     (IX)
PAR1:    LD       (ETYP), HL
CALL    DPEI
CALL    Z17L             ;17. Zeile loeschen
LD       DE, 0180BH
CALL    KOMR             ;Komma schreiben
;
TST2:   LD       BC, 00006H
LD       DE, (RAMA)
CALL    TBIT             ;neue Position ermitteln
BIT     1, M
SET     1, M
JRZ     PAR3
RES     0, M             ;Korrektur bei Typangabe forts.
JMP     TST1
PAR3:   LD       HL, TXRA      ;Text: 'RAM- Anfangsadresse ....'
CALL    TAUS
LD       DE, (RAMA)
CALL    HDS              ;wenn schon vorhanden, ausschrei-
                          ;auf BS
;
LD       DE, 0180CH
CALL    Z4DE             ;Position
LD       (RAMA), HL      ;merken
JRC     TST2
CALL    DPEI
LD       A, B
CMP     004H             ;4- stellige Zahl angegeben ?
JRZ     PAR4
LD       HL, (ADPU)
LD       M, 019H         ;Rest der Zeile loeschen
LD       BC, 00004H      ;akt. Pos. um 4 decrementieren
OR      A
SBC     HL, BC
LD       (ADPU), HL

```

0  
PRF1:  
PAR4:

```
LD IX,PAR3 ;Sprungadresse nach Fehlerben.
JMP PARF
CALL Z17L ;17. Zeile loeschen
LD DE,01810H
CALL KOMR ;Komma schreiben
LD HL,FLAG
RES 4,M
```

0  
1  
TST3:

```
JR PAR5
LD BC,0000CH ;aktuelle Pos. ermitteln
LD DE,(RAME)
CALL TBIT
BIT 2,M
SET 2,M
JRZ PAR5
RES 1,M ;weiter vorn korrigieren
```

0  
1  
PAR5:

```
JMP TST2
CALL Z17L ;17. Zeile loeschen
LD HL,TXRE ;Text: 'RAM- Endadresse ...'
CALL TAUS
LD DE,(RAME)
CALL HDS ;ggf. ausschreiben
LD DE,01811H ;Pos. fuer naechste Eingabe
CALL Z4DE
LD (RAME),HL ;RAM- Endadresse merken
JRZ TST3
CALL DPEI
LD A,B
CMP 004H
JRZ PAR6
LD HL,(ADPU)
LD M,019H
LD BC,00004H
OR A
SBC HL,BC
LD (ADPU),HL
LD IX,PAR5+03H ;Sprungadresse nach Fehlerbehandl.
JMP PARF
```

PAR6:

```
LD A,(ETYP) ;EPROM- Typ
LD HL,(RAME) ;RAM- Endadresse
LD BC,(RAMA) ;RAM- Anfangsadresse
SBC HL,BC ;Laenge in Byte
CMP 008H ;2708 ?
JRNZ PT16
```

PTEN:

```
LD A,003H ;L (= 1k ?)
SUB H
JR PAR7
```

PT16:

```
CMP 016H ;2716 ?
JRNZ PT32
```

PT32:

```
LD A,007H ;L (= 2k ?)
JR PTEN
LD A,00FH ;2732 --> L (= 4k ?)
JR PTEN
```

PAR7:

```
LD (BYTE),HL ;Laenge abspeichern
JRNC PAR8
```

PF01:

```
LD HL,(KOPU) ;zur Korrektur der fehlerhaften
; Eingabe neue Position ermitteln
```

```
LD BC,00006H
ADD HL,BC
LD (ADPU),HL
JMP PARF
```

TST4:

```
LD BC,00011H
LD DE,(RAME)
CALL TBIT ;akt. Position ermitteln
BIT 3,M
```

```

SET          3,M
JRZ          PAR8
RES          2,M           ;weiter vorn korrigieren
RES          4,M
JMP          TST3
PAR8:       CALL        Z17L           ;17. Zeile loeschen
LD          HL, TXRO       ;Text: 'ROM- Anfangsadresse ...'
CALL        TAUS
LD          DE, 01815H
CALL        KOMR           ;Komma schreiben
LD          DE, (ROMA)
CALL        HDS           ;ggf. auf BS schreiben
LD          DE, 01816H     ;neuen Eingabeposition
CALL        Z4DE
LD          (ROMA), HL
JRC         TST4
CALL        DPEI
LD          A, B
CMP         004H
JRZ         PAR9
LD          HL, (ADPU)     ;Position zur Fehlerben. korr.
LD          BC, 00004H
OR          A
SBC        HL, BC
LD          (ADPU), HL
LD          IX, PAR8+03H
JMP        PARF
PAR9:       LD          HL, (BYTE)     ;Laenge
LD          DE, (ROMA)           ;ROM- Anfangsadresse
ADD        HL, DE
LD          (ROME), HL         ;ROM- Endadresse
LD          A, (ETYP)          ;EPROM- Typ
CMP         008H              ;2708 ?
JRNZ       PD16
LD          A, 003H           ;ROME ( 0400H ?
PR10:      SUB          H
JR         PA10
PD16:      CMP         016H       ;2716 ?
JRNZ       PD32
LD          A, 007H           ;ROME ( 0800H ?
JR         PR10
PD32:      LD          A, 00FH     ;2732 --> ROME ( 1000H ?
JR         PR10
PA10:      LD          IX, PAR3     ;Sprungadresse nach Fehlerben.
JPC        PF01
LD          DE, 0181AH
CALL        KOMR           ;Komma schreiben
LD          HL, (ROME)
LD          DE, 0181BH
CALL        Z4HA           ;ROME auf BS schreiben
LD          DE, 0181FH
CALL        KOMR           ;Komma schreiben
FORT:      LD          HL, TXPA     ;Text: 'P: Progr. L: Lesen ...'
CALL        TAUS
LD          BC, 00006H
LD          HL, TXPA
LD          DE, 00000H
TST5:     CALL        TAB           ;Tastaturabfrage
JRZ        TST5
LD          IX, PAR6
CMP         013H           ;'OFF'
JPZ        PARF
LD          (PTYP), A
CMP         050H           ;P = Progr. ?
LD          E, 003H       ;Position im Text

```

```

JRZ PA11
CMP 04CH ;L = Lesen ?
LD E,00EH
JRZ PA11
CMP 054H ;T = Test ?
LD E,016H
JRZ PA11
CMP 056H ;V = Vergl. ?
JRNZ TST5
LD E,021H
PA11: ADD HL,DE
LD DE,01820H
CALL OUT ;angewandte Funktion auf BS
CALL Z17L ;17. Zeile loeschen
PROG: CALL EIIN ;PIO's inaktivieren
LD A,(ETYP)
CMP 008H ;2708 ?
LD HL,ST08 ;Steuerworte fuer 2708
JRZ PR01
CMP 016H ;2716 ?
LD HL,ST16 ;Steuerworte fuer 2716
JRZ PR01
LD HL,ST32 ;2732 --> Steuerworte fuer 2732
PR01: LD (STNT),HL ;merken
LD HL,00000H ;Fehlerzaehler initialisieren
LD (FEHZ),HL
LD HL,(RAMA) ;RAM- Anfangsadresse
LD DE,(ROMA) ;ROM- Anfangsadresse
LD BC,(BYTE) ;Laenge
INC BC
LD A,(PTYP)
CMP 050H ;P ?
JPZ PR02
CMP 04CH ;L ?
JRZ READ
CMP 054H ;T ?
JRZ TEST
;
;***** Funktion: Vergleich *****
;
VGL: CALL EIN ;Einlesen eines Bytes vom EPROM
CMP M ;(RAM) = (ROM) ?
JRZ VGL2
CALL FINC ;nein, Fehlerzaehler incr.
VGL2: CALL AINC ;Adressen incrementieren
JRNZ VGL ;weiter bis Endadresse
JMP ENDE ;zur Endeben.
;
;***** Funktion: Test auf Programmierfaehigkeit *****
;
TEST: CALL TETR ;Testroutine
JMP ENDE ;Endebehandlung
;
;***** Funktion: Lesen *****
;
READ: CALL EIN ;Lesen eines Bytes vom EPROM
LD M,A ;im RAM eintragen
CALL AINC ;Adressen incrementieren
JRNZ READ
LD A,054H ;anschliessend Vergleich
LD (PTYP),A
JMP PROG
;
;***** Funktion: Programmierung eines EPROM's *****
;

```

6

```

PROG: CALL TESR ; Testroutine aufrufen
      LD HL, (FEHZ)
      LD A, H
      OR L
      JPNZ ENDE ; bei Fehler zur Endeben.
      LD A, (ETYP)
      CMP 008H ; 2708 progr. ?
      JRNZ PR16 ; ja, 100 Programmierschleifen
      LD A, 099H ; vorsehen
;
POBA: CALL EIIN ; Eingabeinitialisierung
      PUSH AF
POBI: LD A, 076H ; 26 V einschalten
      OUT EP+02H
      CALL OUT1 ; Adresse und Daten an EPROM
      LD A, 007H ; CTC als Zeitgeber ohne Int.
      OUT EP+08H
      LD A, 0FFH ; Zeitbasis 1ms
      OUT EP+08H
IOBN: IN EP+08H
      CMP 0E8H
      JRNC IOBN ; 150 microsec. warten
      LD A, (BTWT) ; Steuerwort fuer Progr. 2708
      OUT EP+02H ; Programmierimpuls ein
IOBM: IN EP+08H
      CMP 080H
      JRNC IOBM ; 900 microsec. ein
      LD A, 07FH ; Programmierimpuls aus
      OUT EP+02H
      CALL AINC ; Adressen incr.
      JRNZ POBI ; etwas warten
      LD C, 010H
      CALL ZEIT
      CALL FTST ; auf Fehler testen
      LD DE, (FEHZ)
      LD A, D
      OR E
      JRNZ POBB
      LD HL, (FLAG)
      BIT 6, A
      SET 6, A
      JRNZ POBC ; Fehlerzahl 0000 nur einmal
; schreiben
POBB: EX DE, HL
      LD DE, 01728H
      CALL Z4HA
POBC: POP AF
      SUB 001H ; Schleifenzaeher decr.
      DAA
      PUSH AF
      LD HL, TPUF
      PUSH HL
      CALL HBS
      LD BC, 00002H
      POP HL
      LD DE, 01720H ; Schleifenzaeher auf BS
      CALL OUT
      POP AF
      JRNZ POBA ; Progr. bis Schleifenzaeher = 0
      JMP ENDE ; zur Endeben.
; 2716 pro
PR16: CMP 32H
      CPI 0
      LD HL, 00000H ; Anfangsfehlerzahl auf BS
      LD DE, 01728H

```

2708

nein

5FDA  
60B3

2716 pro  
PR16: CMP  
CPI

PR32

```

CALL      Z4HA
CALL      TEBR                ;Testroutine
LD        HL,(FEHZ)
LD        A,H
OR        L
JRNZ     ENDE                ;bei Fehler zur Endeben.
CALL      EIIN                ;Eingabeinitialisierung
P16P:    LD        A,07FH      ;24 V ein
OUT       EPR+02H
CALL     OUT1                ;Byte und Adresse an EPROM
PUSH     BC
LD        BC,00003H          ;etwas warten
CALL     ZEIT
LD        A,(STWT)           ;Steuerwort 2716
OUT      EPR+02H            ;Programmierimpuls ein
LD        B,002H
IN50:    LD        A,027H     ;50 ms lang (2 x 25 ms )
OUT      EPR+08H            ;CTC als Zeitgeber ohne Int.
LD        A,0FFH            ;Zeitkonst. 256
OUT      EPR+08H
IN58:    IN        EPR+08H
        ,
        CMP       02BH
        JRNC     IN58
        DJNZ    IN50        ;50 ms Schleife
LD        A,07FH            ;Programmierimpuls aus
OUT      EPR+02H
LD        BC,00003H        ;etwas warten
CALL     ZEIT
POP      BC
LD        A,04FH
OUT      EPR+07H
CALL     EIN
CMP      M                  ;Vergleich (RAM)(-)(ROM)
PUSH     HL
PUSH     DE
JRZ      P16R
PUSH     BC
CALL     FINC                ;Fehlerzaehler incrementieren
LD        DE,01726H
CALL     Z4HA                ;und ausschreiben
POP      BC
P16R:    PUSH     BC
LD        H,B
LD        L,C
LD        DE,0172EH
CALL     Z4HA                ;Bytezaehler ausschreiben
POP      BC
POP      DE
POP      HL
CALL     AINC                ;Adressen incr.
JRNZ     P16P
ENDE:    LD        HL, TXOK    ;Text: 'FERTIG !'
LD        DE,01710H
LD        BC,0000AH
CALL     OUT
LD        HL, TXFE          ;Text: 'Fehler'
LD        E,021H
CALL     OUT
LD        HL,(FEHZ)
LD        DE,01728H
CALL     Z4HA                ;Fehlerzahl ausschreiben
CALL     EIIN                ;Eingabeinitialisierung
LD        HL,00000H
LD        A,(ETYP)
CMP      08H

```

(8)

```

LD      BC,0400H
JRZ    CRC0
CMP    16H
LD      B,08H
JRZ    CRC0
LD      B,10H
CRC0:  LD      DE,0FFFFH
CRC1:  PUSH   DE
EX      DE,HL
CALL   EIN      ;Byte einlesen
EX      DE,HL
POP    DE
XOR    D
LD      D,A
RRCA
RRCA
RRCA
RRCA
AND    00FH
XOR    D
LD      D,A
RRCA
RRCA
RRCA
PUSH  AF
AND    01FH
XOR    E
LD      E,A
POP    AF
PUSH  AF
RRCA
AND    0F0H
XOR    E
LD      E,A
POP    AF
AND    0E0H
XOR    D
LD      D,E
LD      E,A
INC   HL      ;naechste Adresse
DEC   BC
LD      A,B
OR     C
JRNZ  CRC1    ;weiter bis Bytezaehler = X0000
CALL  HDS
LD      HL,TCRC ;Text: 'CRC:'
LD      DE,01738H
LD      BC,00004H
CALL  OUT
LD      HL,TPUF
LD      E,03DH
CALL  OUT      ;CRC- Summe auf BS schreiben
CALL  EINI     ;PIO's inaktiv
END2:  LD      C,070H ;etwas warten
CALL  ZEIT
LD      HL,PIEP ;dann 50ms 'Pilleep'
LD      BC,00001H
CALL  BFE
CALL  WBN
LD      C,0C0H ;und wieder etwas warten
CALL  ZEIT
LD      A,(LZVT) ;letztes Zeichen der Tastatur
CMP    011H     ;'ENTER'
JRNZ  END1
LD      HL,TPUF

```



9

```

LD      M,019H      ;ein Stueck der untersten Zeile
                        loeschen
LD      DE,01820H
CALL    BFE
LD      BC,00001H
CALL    WBN
CALL    Z17L      ;17. Zeile loeschen
JMP     FORT      ;im Eingabemodus fortsetzen
END1:   CMP         013H      ;'OFF'
JRNZ    ENDZ
LD      HL,(KQPU)   ;ja, Programm verlassen
LD      DE,(ADPU)
POP     DE
POP     IX
POP     IY
LD      A,000H
XOR     A          ;Cy = 0
RET

```

\*\*\*\*\* U N T E R P R O G R A M M E \*\*\*\*\*

```

Z4DE:   LD      BC,00005H      ;4 Ziffern und Abschlusstaste
        PUSH    DE
        LD      HL,FLAG      ;Fehlerbit
        BIT     7,M
        LD      HL,TPUF
        PUSH    HL
        JRNZ    Z4D0
Z4D0:   LD      M,000H      ;Textpuffer loeschen
        CALL    BFE
        LD      A,048H
        LD      (IY+08H),000H ;Textpufferkursor auf X00
        CALL    LIY
        CALL    CON      ;Aufruf Console
        LD      A,(LZVT)    ;letztes Zeichen der Tastatur
        CMP     0F6H      ;'cV' ?
        JPZ     TVWR      ;ja
        POP     DE
        LD      A,084H
        CALL    ZLE      ;Zahl von Textpuffer lesen
        LD      B,A        ;Anzahl der Ziffern in B
        JRC     Z4D1
Z4D1:   LD      B,004H
        POP     DE
        LD      A,(LZVT)
        CMP     011H      ;Abschluss mit 'ENTER' ?
        RZ
        CMP     013H      ;oder mit 'OFF' ?
        SCF      ;dann Cy = 1
        RZ
        JR      Z4DE
TVWR:   POP     DE      ;Versionsnummer auf BS schreiben
        POP     DE
        POP     DE
        LD      HL,TVWR    ;Text: 'VERSION 3.1.'
        CALL    TAUS
TST0:   LD      A,(LZVT)
        CMP     013H      ;RET mit 'OFF'
        JRNZ    TST0
        POP     DE
        POP     IX
        POP     IY

```

(10)

```
XOR      A
RET

;
24HA:   PUSH    DE                ;4- stellige Hexazahl auf BS
        EX      DE,HL
        CALL   HDS                ;in TPUF eintragen
        POP    DE
        LD     HL,TPUF
        LD     BC,00004H
        CALL   OUT                ;und nun ausschreiben
        RET

;
KOWR:   LD      HL,TPUF            ;Komma auf (DE) schreiben
        LD     BC,00001H
        LD     N,02CH
        CALL   OUT
        LD     HL,(ADPU)          ;akt. Position neu vermerken
        INC    HL
        LD     (ADPU),HL
        RET

;
TAUS:   PUSH    DE                ;Text ab Zeiger (HL) auf BS
        PUSH   BC
        LD     DE,01720H
        LD     BC,00028H          ;max. 40 Zeichen
        CALL   OUT
        POP    BC
        POP    DE
        RET

;
Z17L:   PUSH    AF                ;17. Zeile loeschen
        PUSH   BC
        PUSH   DE
        PUSH   HL
        LD     HL,TPUF
        LD     BC,00001H
        LD     N,019H            ;Stwort., fuer: Rest der Zeile loe-
                                   schen
        LD     DE,01700H
        CALL   BFE
        CALL   WBN
        POP    HL
        POP    DE
        POP    BC
        POP    AF
        RET

;
DPEI:   PUSH    BC                ;Zeiger akt. Position aktualisieren
        LD     DE,(ADPU)
        LD     HL,TPUF
        LD     BC,00004H
        LDIR
        LD     (ADPU),DE
        POP    BC
        RET

;
HDS:   PUSH    HL                ;Zahl aus DE auf TPUF schreiben
        LD     HL,TPUF
        LD     A,D
        CALL   HBS
        LD     A,E
        CALL   HBS
        POP    HL
        RET
```

(11)

```

FINC:  PUSH    HL                ;Fehlerzaehler incrementieren
        LD      HL,(FEHZ)
        INC    HL
        LD      (FEHZ),HL
        POP    HL
        RET

;
AINC:  INC     HL                ;Adressen incrementieren
        INC    DE
        DEC    BC                ;Bytezaehler decreamentieren
        LD     A,B
        OR    C
        RET

;
TSR1:  CALL    EIIN              ;Testroutine / Eingabeinitial.
TSR1:  CALL    EIN               ;Byte vom EPROM lesen
        XOR    M
        AND    M                ;Byte programmierfaehig ?
        JNZ   TSR2
        CALL   FINC              ;nein, Fehlerzaehler incr.
TSR2:  CALL   AINC               ;Adressen incr.
        JRNZ  TSR1
        RET

;
TBIT:  LD      HL,FLAG           ;Korrekturbehandlung
        BIT   4,M
        SET   4,M
        SET   7,M
        JNZ   TRET
        CALL  HDS                ;ggf. Wert aus DE auf TPUF
TRET:  PUSH    HL
        LD     HL,(KOPU)         ;Zeiger akt. Pos. aktualisieren
        ADD   HL,BC
        LD     (ADPU),HL
        POP    HL

;
EIN:   LD      A,E              ;Einlesen eines Bytes vom EPROM,
;                                     L- Teil ROM- Adresse an EPROM
;                                     ;wird im Komplement angelegt
        CPL
        OUT   EPR
        LD    A,D                ;H- Teil
        CPL
        OUT   EPR+04H
        LD    A,(STWT+01H)       ;Steuerwort fuer Lesen
        OUT   EPR+02H
        IN   EPR+06H            ;Byte in A lesen
        RET

;
EINI:  LD      A,04FH           ;PIO's inaktivieren
        OUT   EPR+01H
        OUT   EPR+03H
        OUT   EPR+05H
        OUT   EPR+07H
        LD    A,0FFH
        OUT   EPR
        OUT   EPR+02H
        OUT   EPR+04H
        XOR   A
        OUT   EPR+06H
        RET

;
EIIN:  PUSH    AF                ;Eingabeinitialisierung
        LD     A,00FH
        OUT   EPR+01H

```

116

```

OUT      EPR+03H
OUT      EPR+05H
LD       A,04FH
OUT      EPR+07H
ANFA:    LD      DE,(ROMA)      ;ROM- Anfangsadresse
         LD      HL,(RAMA)     ;RAM- Anfangsadresse
         LD      BC,(BYTE)    ;Laenge
         INC     BC
         POP     AF
         RET

ZEIT:    DJNZ   ZEIT          ;Zeitschleife
         DEC    C
         JRNZ  ZEIT
         RET

OUT1:    LD      A,00FH        ;Byte und Adresse an EPROM
         OUT    EPR+07H
         LD      A,M
         OUT    EPR+06H      ;Byte ausgeben
         LD      A,E
         CPL
         OUT    EPR          ;L- Teil Adresse
         LD      A,D
         CPL
         OUT    EPR+04H      ;H- Teil Adresse
         RET

FTST:    PUSH   AF           ;Fehlerst nach Progn.
         PUSH   BC
         PUSH   DE
         PUSH   HL
         LD     HL,00000H
         LD     (FEHZ),HL
         CALL  EIIN         ;Eingabeinitialisierung
FTS1:    CALL  EIN          ;Byte einlesen
         CMP   M            ;(RAM) = (ROM) ?
         JRZ  FTS2
         PUSH  HL           ;nein, Fehlerzaehler incr.
         CALL  FINE
         POP   HL
FTS2:    CALL  AINC         ;Adressen incr.
         JRNZ FTS1
         POP   HL
         POP   DE
         POP   BC
         POP   AF
         RET

```

7A48 7B75

\*\*\*\*\* A R B E I T S B E R E I C H \*\*\*\*\*

```

ADPU:    EQU    0F911H
KOPU:    EQU    0F913H
TRUF:    EQU    0F915H
FLAG:    EQU    0F91BH
ETYP:    EQU    0F91CH
RAMA:    EQU    0F91EH
RAME:    EQU    0F920H
ROMA:    EQU    0F922H
ROME:    EQU    0F924H
BYTE:    EQU    0F926H
PTYP:    EQU    0F928H

```

BTWT: EQU 0F929H  
FEHZ: EQU 0F92BH

\*\*\*\*\* T E X T T A B E L L E \*\*\*\*\*

TXFE: DM 'Fehler !'  
DB OFFH  
TXTA: DM 'Typ angeben: 2708 - 2716' -2732  
DB OFFH  
TXRA: DM 'RAM- Anfangsadresse angeben !'  
DB OFFH  
TXRE: DM 'RAM- Endadresse angeben !'  
DB OFFH  
TXRD: DM 'EPROM- Anfangsadresse angeben !'  
DB OFFH  
TXPA: DM 'P: Progr. L: Lesen T: Test V: Vergl.'  
DB OFFH  
TXOK: DM 'FERTIG !'  
PIEP: DB 007H  
DB OFFH  
TCRC: DM 'CRC:'  
DB OFFH  
TVER: DM 'VERSION: 3.1.'  
DB 019H  
DB 007H  
DB OFFH

\*\*\*\*\* E N D E \*\*\* E N D E \*\*\* E N D E \*\*\* E N D E \*\*\*\*\*