```
;
;
;       EPROM-Programmierung  MC 80.3x
;       ==============================
;       ab Version 3.0.
;
;
;
;****** SW-Service                              Tel:
;       ----------
;Fassung 11/85  Bearb.: A. Schimpf / Gera 58-4190
;Fassung 02/90
;
;
;****** VEREINBARUNGSTEIL
;       ------------------
; I/O-Adresse des Plattine
EPR:    EQU 000E0H              ;
;
;UP's aus Betriebssystem
OUT:    EQU 00BBEH              ;Text auf Display schreiben
MBN:    EQU 00BC1H              ;Text auf Display schreiben
TAS:    EQU 00BCDH              ;Tastatur abfragen
ZLE:    EQU 00BD0H              ;Zahl von Texptuffer lesen
BFE:    EQU 00BD6H              ;Bildfenster festlegen
LIY:    EQU 00BE8H              ;IY- Vektor laden
KBS:    EQU 00BEBH              ;Byte auf Textpuffer schr.
CON:    EQU 00BEEH              ;Consolentreiber
;
;AZ des Betriebssystems
LZVT:   EQU 0FB62H
;
; Steuerwort für EPROM-Type
ST08:   EQU 0C870H              ;EIN für 2708
ST16:   EQU 0CA32H              ;EIN für 2716
ST32:   EQU 0CC6CH              ;EIN für 2732
;
;****    Arbeitszellen des Programms
;       ---------------------------
;
ADPU:   EQU     0F911H
KDPU:   EQU     0F913H
TPUF:   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
STMT:   EQU     0F929H
FEHZ:   EQU     0F92BH
;
;
;
;***** H A U P T P R O G R A M M
;      ========================
;
PARA:   PUSH IY
        PUSH IX
;Arbeisbereich initialisieren
```

```
          LD HL,TPUF
          LD B,18H
PI00:     LD M,000H
          INC HL
          DJNZ PI00
          LD A,(DE)
          CMP 0FFH             ;z.Z. nur mit
          JPNZ PART            ;Parameterabfr.
          LD (KOPU),DE         ;Kdo-pufferanf.
          PUSH DE
          EX DE,HL
          LD M,020H
          INC HL
          LD (ADPU),HL         ;akt. Pos.
          JR PAR0
;
;Behandlung im Korr.modus
TST1:     LD HL,FLAG
          BIT 0,M
          SET 0,M
          SET 7,M
          PUSH AF
          LD HL,(KOPU)         ;akt. Pos. erm.
          INC HL
          LD (ADPU),HL
          PUSH HL
          PUSH DE
;EPROM-Typ auf Display schreiben
          LD DE,(ETYP)
          CALL HDS
          POP DE
          POP HL
          POP AF
          JRZ PAR0
          LD HL,(KOPU)         ;fertig
          LD M,0FFH
          POP DE
PART:     POP IX
          POP IY
          OR A
          RET
;
;BIS: "EPROM-Typ angeben ..."
PAR0:     LD HL,TXTA
          CALL TAUS            ;ausschreiben
          LD DE,01807H         ;Displaypos.
          CALL Z4DE            ;4-st-hex., eing.
          JRC TST1             ;Fehler
          LD A,H
          CMP 027H
          LD IX,PAR0           ;Sprungadr.
          JRNZ PARF            ;bei Fehler
          LD A,L
          CMP 008H             ;2708 ?
          JRZ PAR1
          CMP 016H             ;2716 ?
          JRZ PAR1
          CMP 032H             ;2732 ?
          JRZ PAR1
PARF:     LD DE,01710H         ;Position f.
          LD HL,FLAG           ;Fehlerausschr.
          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 lö.
          LD DE,0180BH
          CALL KOWR           ;Komma schreiben
          JR PAR3
;
;RAM-Anfangsadresse eingeben
TST2:     LD BC,00006H
          LD DE,(RAMA)
          CALL TBIT           ;neue Pos. erm.
          BIT 1,M
          SET 1,M
          JRZ PAR3
          RES 0,M             ;Korrektur bei.
          JMP TST1            ;falscher Type
;
;BIS: "RAM-Anfangsadresse ..."
PAR3:     LD HL,TXRA
          CALL TAUS
          LD DE,(RAMA)
          CALL HDS
          LD DE,0180CH        ;Position
          CALL Z4DE
          LD (RAMA),HL        ;merken
          JRC TST2
          CALL DPEI
          LD A,B
          CMP 004H            ;Zahl 4-stellig?
          JRZ PAR4
          LD HL,(ADPU)
          LD M,019H           ;Rest der Z. lö.
          LD BC,00004H        ;akt. Pos. um 4 dec.
          OR A
          SBC HL,BC
          LD (ADPU),HL
          LD IX,PAR3          ;Sprungadr. ---> Fehlerbeh.
PRF1:     JMP PARF
;
PAR4:     CALL Z17L           ;17. Z. lö.
          LD DE,01810H
          CALL KOWR           ;Komma schreiben
          LD HL,FLAG
          RES 4,M
          JR PAR5
;
;RAM-Endadresse ermitteln
TST3:     LD BC,0000CH        ;akt. Pos. erm.
          LD DE,(RAME)
          CALL TBIT
          BIT 2,M
          SET 2,M
          JRZ PAR5
          RES 1,M             ;Fehler: Korr.
          JMP TST2            ;anspringen
;
;BIS: "RAM-Endadresse ..."
PAR5:     CALL Z17L           ;17. Zeile lö.
          LD HL,TXRE
          CALL TAUS
          LD DE,(RAME)
          CALL HDS            ;ggf. ausschr.
          LD DE,01811H        ;Pos. f. n. Eingabe
```

```
        CALL Z4DE
        LD (RAME),HL        ;RAM-Endadr. merken
        JRC 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      ;z. Fehlerbeh.
        JMP PARF            ;springen
;
PAR6:   LD A,(ETYP)         ;EPROM- Typ
        LD HL,(RAME)        ;RAM- Endadr.
        LD BC,(RAMA)        ;RAM- Anfangsadr.
        SBC HL,BC           ;Länge in Byte
        CMP 008H            ;2708 ?
        JRNZ PT16
        LD A,003H           ;L (= 1k ?
PTEW:   SUB H
        JR PAR7
PT16:   CMP 016H            ;2716 ?
        JRNZ PT32
        LD A,007H           ;L (= 2k ?
        JR PTEW
PT32:   LD A,00FH           ;L (= 4k ?
        JR PTEW
;
PAR7:   LD (BYTE),HL        ;Länge absp.
        JRNC PAR8
        LD IX,PAR3
;
;z. Korr. d. fehlerh. Eingabe neue Pos. erm.
PF01:   LD HL,(KOPU)
        LD BC,00006H
        ADD HL,BC
        LD (ADPU),HL
        JMP PARF
;
TST4:   LD BC,00011H
        LD DE,(RAME)
        CALL TBIT           ;akt. Position erm.
        BIT 3,M
        SET 3,M
        JRZ PAR8
        RES 2,M             ;weiter vorn korr.
        RES 4,M
        JMP TST3
;
;BIS: "ROM-Anfangsadresse ..."
PAR8:   CALL Z17L           ;17. Zeile lö.
        LD HL,TXRO
        CALL TAUS
        LD DE,01815H
        CALL KOWR           ;Komma schreiben
        LD DE,(ROMA)
        CALL HDS            ;ggf. auf BIS schr.
        LD DE,01816H        ;neuen Eingabepos.
        CALL Z4DE
        LD (ROMA),HL
        JRC TST4
```

```
        CALL DPEI
        LD A,B
        CMP 004H
        JRZ PAR9
        LD HL,(ADPU)        ;Pos. zur
        LD BC,00004H        ;Fehlerbeh.
        OR A                ;wird
        SBC HL,BC           ;korregiert
        LD (ADPU),HL        ;merken
        LD IX,PAR8+03H
        JMP PARF

;
PAR9:   LD HL,(BYTE)        ;Länge
        LD DE,(ROMA)        ;ROM- Anfangsadr.
        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           ;ROME < 1000H ?
        JR PR10

;
PA10:   LD IX,PAR3          ;Sprungadr. nach
        JPC PF01            ;Fehlerbeh.
        LD DE,0181AH
        CALL KOMR           ;Komma schreiben
        LD HL,(ROME)
        LD DE,0181BH
        CALL Z4HA           ;ROME auf BIS schreiben
        LD DE,0181FH
        CALL KOMR           ;Komma schreiben
FORT:   LD HL,TXPA          ;'P: Progr.  L: Lesen ...'
        CALL TAUS
        LD BC,00006H
        LD HL,TXPA
        LD DE,00000H

;
;Tastaturabfrage
TST5:   CALL TAS
        JRZ TST5
        LD IX,PAR6
        CMP 013H            ;'OFF'?
        JPZ PARF            ;ja
        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,018H
        JRZ PA11
        CMP 056H            ;V = Vergl. ?
        JRNZ TST5
        LD E,021H

;
;angewählte Funktion auf BIS ausgeben
```

```
PA11:   ADD HL,DE
        LD DE,01820H
        CALL OUT
        CALL Z17L           ;17. Zeile 16.
PROG:   CALL EIIN           ;PIO's inaktiv.
        LD A,(ETYP)
        CMP 008H            ;2708 ?
        LD HL,ST08          ;STW f. 2708
        JRZ PRO1
        CMP 016H            ;2716 ?
        LD HL,ST16          ;STW f. 2716
        JRZ PRO1
        LD HL,ST32          ;STW f. 2732
PRO1:   LD (STWT),HL        ;merken
        LD HL,00000H        ;Fehlerz. init.
        LD (FEHZ),HL
        LD HL,(RAMA)        ;RAM- Anfangsadr.
        LD DE,(ROMA)        ;ROM- Anfangsadr.
        LD BC,(BYTE)        ;Länge
        INC BC
        LD A,(PTYP)
        CMP 050H            ;P ?
        JPZ PRO2
        CMP 04CH            ;L ?
        JRZ READ
        CMP 054H            ;T ?
        JRZ TEST
;
;
;****** Vergleich
;       ----------
;
VGL:    CALL EIN            ;1 By. vom EPROM einl.
        CMP M               ;(RAM)==(ROM) ?
        JRZ VGL2            ;ja
        CALL FINC           ;Fehlerz. inc.
VGL2:   CALL AINC           ;Adr. inc.
        JRNZ VGL            ;weiter b. Endadr.
        JMP ENDE
;
;
;****** Test auf Programmierfähigkeit
;       ----------------------------------
;
TEST:   CALL TESR           ;Testroutine
        JMP ENDE
;
;
;****** Lesen
;       -----
;
READ:   CALL EIN            ;1 Bytes vom EPROM einl.
        LD M,A              ;im RAM eintr.
        CALL AINC           ;Adr. inc.
        JRNZ READ
        LD A,056H           ;anschl. Vergleich
        LD (PTYP),A
        JMP PROG
;
;
;****** Programmierung eines EPROM's
;       ----------------------------------
;
PRO2:   CALL TESR           ;Testroutine
        LD HL,(FEHZ)
```

```
          LD A,H
          OR L                    ;Fehler?
          JPNZ ENDE               ;ja
          LD A,(ETYP)
          CMP 008H                ;2708 progr. ?
          JPNZ PR16               ;nein
;                          Typ
; Programmierung 2708
          LD A,099H               ;100 Schleifen
P08A:     CALL EIIN               ;Eingabe-INIT
          PUSH AF
P08I:     LD        A,078H                ;26 V einschalten
          OUT       EPR+02H
          CALL      OUT1                  ;Adresse und Daten an EPROM
          LD        A,007H                ;CTC als Zeitgeber ohne Int.
          OUT       EPR+08H
          LD        A,0FFH                ;Zeitbasis 1ms
          OUT       EPR+08H
I08N:     IN        EPR+08H
          CMP       0E8H
          JRNC      I08N                  ;150 microsec. warten
          LD        A,(STWT)              ;Steuerwort fuer Progr. 2708
          OUT EPR+02H             ;Progr.imp. ein
I08M:     IN EPR+08H
          CMP 080H
          JRNC I08M               ;0.9 ms ein
          LD A,07FH               ;Progr.imp. aus
          OUT       EPR+02H
          CALL      AINC                  ;Adressen incr.
          JRNZ      P08I
          LD        C,010H                ;etwas warten
          CALL      ZEIT
          CALL      FTST                  ;auf Fehler testen
          LD        DE,(FEHZ)
          LD        A,D
          OR        E
          JRNZ      P08B
          LD        HL,(FLAG)
          BIT       6,A                   Fehlerzahl 0000
          SET       6,A                   nur 1x
          JRNZ      P08C                  ;Fehlerzahl 0000 nur einmal
;                                         schreiben
P08B:     EX        DE,HL
          LD        DE,01720H
          CALL      Z4HA
P08C:     POP AF
          SUB 001H                ;Schleifenz. dec.
          DAA
          PUSH AF
          LD        HL,TPUF
          PUSH      HL
          CALL      HBS
          LD        BC,00002H
          POP       HL
;Schleifenzähler auf BiS ausgeben
          LD DE,0172DH
          CALL OUT
          POP AF
          JRNZ P08A               ;Schleifenz. = 0
          JMP ENDE                ;ja
;
;                          Typen
;Programmierung 2716/2732
PR16:     LD HL,00000H            ;Anf.fehlerz.
          LD DE,01728H            ;auf BiS
```

```
          CALL Z4HA           ;ausgeben
          CALL TESR           ;Testroutine
          LD HL,(FEHZ)
          LD A,H
          OR L                ;Fehler ?
          JRNZ ENDE           ;ja
          CALL EIIN           ;Eing.-INIT

;
P16P:     LD A,07FH           ;24 V ein
          OUT EPR+02H
;Byte und Adresse an EPROM
          CALL OUT1
          PUSH BC
          CALL ZSL
          LD A,(STMT)         ;STM 2716/2732
          OUT EPR+02H         ;Progr.imp. ein
          LD B,002H
IN50:     LD A,027H           ;50 ms (2x25)
          OUT EPR+08H         ;CTC: ZG ohne IR
          LD A,0FFH           ;ZK: 256
          OUT EPR+08H
INE8:     IN EPR+08H
          CMP 025H
          JRNC INE8
          DJNZ IN50           ;50 ms ZSL
          LD A,07FH           ;Progr.imp. aus
          OUT EPR+02H
          CALL ZSL
          POP BC
          LD A,04FH
          OUT EPR+07H
          CALL EIN
          CMP M               ;(RAM)==(ROM)?
          PUSH HL
          PUSH DE
          JRZ P16R            ;ja
          PUSH BC
          CALL FINC           ;Fehlerz. inc.
          LD DE,01728H        ;auf BIS
          CALL Z4HA           ;ausschreiben
          POP BC
P16R:     PUSH BC
          LD H,B
          LD L,C
          LD DE,0172EH        ;auf BIS:
          CALL Z4HA           ;Bytezähler
          POP BC
          POP DE
          POP HL
          CALL AINC           ;Adressen inc.
          JRNZ P16P

;
;BIS: "FERTIG!"
ENDE:     LD HL,TXOK
          LD DE,01710H
          LD BC,0000AH
          CALL OUT
          LD HL,TXFE          ;'Fehler'
          LD E,021H
          CALL OUT
          LD HL,(FEHZ)
          LD DE,01728H        ;Fehlerzahl
          CALL Z4HA           ;ausschreiben
;Eingabeinitialisierung
          CALL EIIN
```

```
          LD HL,00000H
          LD        A,(ETYP)
          CMP 08H
          LD BC,0400H
          JRZ CRC0
          CMP 18H
          LD B,08H
          JRZ CRC0
          LD B,10H
;
;
; CRC-Summenbildung
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                          ;BC==0000?
          JRNZ CRC1                     ;nein
;
          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 'Pieeep'
          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
            LD      M,019H          ;ein Stueck der untersten Zeile
;                                    loeschen
            LD      DE,0182OH
            CALL    BFE
            LD      BC,00001H
            CALL    WBN
            CALL    Z17L            ;17. Zeile loeschen
            JMP     FORT            ;im Eingabemodus fortsetzen
END1:       CMP     013H            ;'OFF'
            JRNZ    END2
            LD      HL,(XOPU)       ;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
            LD      M,000H          ;Textpuffer loeschen
Z4D0:       CALL    BFE
            LD      A,048H
            LD      (IY+0BH),000H   ;Textpufferkursor auf X00
            CALL    LIY
            CALL    CON             ;Aufruf Console
            LD      A,(LZVT)        ;letztes Zeichen der Tastatur
            CMP     0F6H            ;'cV' ?
            JPZ     TVNR            ;ja
            POP     DE
            LD      A,084H
            CALL    ZLE             ;Zahl von Textpuffer lesen
            LD      B,A             ;Anzahl der Ziffern in B
            JRC     Z4D1
            LD      B,004H
Z4D1:       POP     DE
            LD      A,(LZVT)
            CMP     011H            ;Abschluss mit 'ENTER' ?
            RZ
            CMP     013H            ;oder mit 'OFF' ?
            SCF                     ;dann Cy = 1
            RZ
            JR      Z4DE
TVNR:       POP     DE              ;Versionsnummer auf 85 schreiben
            POP     DE
            POP     DE
```

```
        LD      HL,TVER         ;Text: 'VERSION 3.1.'
        CALL    TAUS
TSTO:   LD      A,(LIVT)
        CMP     013H            ;RET mit 'OFF'
        JRNZ    TSTO
        POP     DE
        POP     IX
        POP     IY
        XOR     A
        RET
;
Z4HA:   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
;
KOMA:   LD      HL,TPUF         ;Komma auf (DE) schreiben
        LD      BC,00001H
        LD      M,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      M,019H          ;Stwrt. fuer: Rest der Zeile loe-
;                                schen
        LD      DE,01700H
        CALL    BFE
        CALL    MBN
        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
;
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 decrementieren
        LD      A,B
        OR      C
        RET
;
TESR:   CALL    EIIN            ;Testroutine / Eingabeinitial.
TSR1:   CALL    EIN             ;Byte vom EPROM lesen
        XOR     M
        AND     M               ;Byte programmierfaehig ?
        JRZ     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
        JRZ     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
        RET
;
EIN:    LD      A,E             ;Einlesen eines Bytes vom EPROM,
;                                L- Teil ROM- Adresse an EPROM
        CPL                     ;wird im Komplement angelegt
        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
        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
;
ZSL:    LD BC,03H
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              ;Fehlertest nach Progr.
        PUSH    BC
        PUSH    DE
        PUSH    HL
        LD      HL,00000H
        LD      (FEHZ),HL       ;Fehlerzaehler init.
        CALL    EIIN            ;Eingabeinitialisierung
FTS1:   CALL    EIN             ;Byte einlesen
        CMP     M               ;(RAM) = (ROM) ?
        JRZ     FTS2
        PUSH    HL              ;nein, Fehlerzaehler incr.
        CALL    FINC
        POP     HL
FTS2:   CALL    AINC            ;Adressen incr.
        JRNZ    FTS1
        POP     HL
        POP     DE
        POP     BC
        POP     AF
        RET
;
;
;****** T E X T T A B E L L E
;       ---------------------
;
;
TXFE:   DM      'Fehler !'
```

```
        DB      0FFH
TXTA:   DM      'Typ angeben: 2708 - 2716 - 2732'
        DB      0FFH
TXRA:   DM      'RAM- Anfangsadresse angeben !'
        DB      0FFH
TXRE:   DM      'RAM- Endadresse angeben !'
        DB      0FFH
TXRO:   DM      'EPROM- Anfangsadresse angeben !'
        DB      0FFH
TXPA:   DM      'P: Progr.  L: Lesen  T: Test  V: Vergl.'
        DB      0FFH
TXOK:   DM      'FERTIG !'
PIEP:   DB      007H
        DB      0FFH
TCRC:   DM      'CRC:'
        DB      0FFH
TVER:   DM      'VERSION: 3.1 2732'
        DB      019H
        DB      007H
        DB      0FFH
        DB 000H
;
;
;
```