

Elektronik IV E - Lösungen zu den Arbeitsblättern

Computergestützte Steuerungstechnik

© 1997/00IED München GmbH
Vertrieb: Ing.-Büro Kammerer GmbH
Schmiedweg 5a
D-85551 Kirchheim
Tel 089 / 904 802 31
Fax 089 / 904 802 33
<http://www.kammerer.net>
email: info@kammerer.net

1 Arbeiten mit dem Programm MONI.EXE

1.1 Die Prozessorfunktionen des Programms MONI.EXE

F1: *Der 16-Bit-Wert 1200 wird in den 16-Bit Akkumulator AX transferiert.*

F2: *Codesegment CS
Instructionpointer IP*

F3: *2300:0100
23100*

F4: *Der Befehl MOV AX,<16-Bit-Konstante> ist ein 3-Byte-Befehl*

F5: *UND-Vernüpfung zwischen 16-Bit-Operanden*

F6: *Der erste Operand steht im 16-Bit-Akku AX*

F7: *Der zweite Operand steht im Speicher, die Adresse steht im Register BX*

F8: *im 16-Bit-Akkumulator AX*

A1: *B5A3*

A2: *5C36*

A3 *1011 0101 1010 0011
0101 1100 0011 0110
0001 0100 0010 0010
1422*

1.2 Speicherfunktionen des Programms MONI.EXE

1.2.1 Übungen zum BIOS-Datenbereich

A1: *00 = 00000000*

nein

nein

nein

nein

nein

nein

nein

nein

A2: *70 = 01110000*

nein

ja

ja

ja

nein

nein

nein

nein

Übereinstimmung liegt vor *ja*

1.2.2 Übungen zum Bildschirm-Speicherbereich

- A3:** Adresse von „P“: *B800:000A*
Inhalt der Adresse: *50 = 01010000*
ASCII-Tabelle, siehe Seite 106
Adresse des Attributs: *B800:000B*
Inhalt des Attributs: *74 = 01110100*
- A4:** Zeichenfarbe: *rot*
Hintergrundfarbe: *weiß*
Blinkbit: *0=aus*
- B4:** Moni verlassen mit *F10*, Ende, Ja
Bildschirmmodus ändern mit *MODE BW80*
Moni erneut starten mit *MONI*
- A5:** Inhalt der Adresse: *7F = 01111111*
- A6:** Zeichenfarbe: *hellweiß*
Hintergrundfarbe: *weiß*
Blinkbit: *0=aus*

2 Grundübungen zum DEBUG-Programm

2.1 Registerfunktionen des DEBUG

2.1.1 Registerstatus

A1: Segment-Adressen, Hex-Code, Mnemonic und ggf Speicherinhalt weichen von den abgebildeten Werten ab.

AX=0000	BX=0000	CX=0000	DX=0000	SP=FFEE	BP=0000	SI=0000	DI=0000
DS=12A7	ES=12A7	SS=12A7	CS=12A7	IP=0100	NV UP	EI PL	NZ NA PO NC
12A7:0100	69		DB		69		

F1: *Alle Datenregister sind auf 0000 gesetzt*

F2: *Alle Segmentregister sind den selben Wert*

F3: *auf den Wert 0100*

2.1.2 Register beeinflussen

A2: <DX> = 0000

B2: 0000
38E1

A3: <AX> = 0000

B3: 0000
0097

2.1.3 Beeinflussung der Flagsbits

A4: Überlauf	NV	0	Null	NZ	0
Richtung	UP	0	Hilfsübertrag	NA	0
Interrupt-Freigabe	EI	1	Parität	PO	0
Minuszeichen	PL	0	Übertrag	NC	0

B5: NV UP EI PL NZ NA PO NC - OV ZR

2.2 Speicherfunktionen des DEBUG

2.2.1 Speicherinhalte feststellen

A1: Der Wert der Segmentadresse kann von der folgenden Darstellung abweichen.

DS: 12A7

ES: 12A7

SS: 12A7

CS: 12A7

B4: Segmentadresse und Speicherinhalt können von der folgenden Darstellung abweichen.

12A7:0100	69 6F 6E 0D 0A 12 55 6E-67 81 6C 74 69 67 65 20	ion...Ung.ltige
12A7:0110	46 75 6E 6B 74 69 6F 6E-36 0D 0A 41 34 00 96 12	Funktion6..A4...
12A7:0120	65 20 76 6F 6E 20 43 4F-4D 31 2C 20 43 4F 4D 32	e von COM1, COM2
12A7:0130	2C 20 43 4F 4D 33 20 6F-64 65 72 20 43 4F 4D 34	, COM3 oder COM4
12A7:0140	20 65 72 66 6F 72 64 65-72 6C 69 63 68 0D 0A 24	erforderlich..\$
12A7:0150	0D 0A 52 65 73 69 64 65-6E 74 65 72 20 54 65 69	..Residenter Teil
12A7:0160	6C 20 76 6F 6E 20 4D 4F-44 45 20 67 65 6C 61 64	l von MODE gelad
12A7:0170	65 6E 0D 0A 1B 0D 0A 55-6E 7A 75 6C 84 73 73 69	en.....Unzul.ssi

F1: *Anzeigen der nächsten 128 Byte Speicherinhalt*

B5: 10 94

10 L 85

F2: FFFF0 = FFFF:0000 FFFFF = FFFF:000F

B7: FFFF:0 F

A3: 09/02/96 = 2.September 1996

2.2.2 Speicherinhalte verändern

B2: 200 20F 58

B3: 200 34 37 31 31

B4: XXXX:0203 31.32

2.2.4 Direktzugriff auf den Bildschirmspeicher

F1: 80 x 25 Byte für die Zeichen

80 x 25 Byte für die Attribute

ergibt 4000 Byte Speicher, umgerechnet in Hex-Wert: 0FA0

Anfangsadresse B800:0000

Endadresse B800:0F9F

B1: Der aktuelle Wert des Datensegmentregisters DS kann von der folgenden Darstellung abweichen.

DS 372F

: B800

F2: Offsetadresse des Zeichens: 0F9E

Offsetadresse des Attributs: 0F9F

ASCII-Code des Zeichens 41

Hex-Wert des Attributs 9C

F3: 0F9E "A" 9C

Die Darstellung ist abhängig vom geladenen Zeichensatz!

F4: Offsetadresse des ersten Zeichens 0050

ASCII-Code des Zeichens "k" 6B

ASCII-Code des Zeichens "Ω" EA

Hexwert der zugehörigen Attribute 2E

F5: 50 6B 2E EA 2E

3 Dateifunktionen des DEBUG

3.1 Erstellen einer Textdatei

F1: *e100 "DIES IST EIN VERSUCH"*

B3: *A:\TEXT1.DAT*

F2: Länge = 20 (dezimal) = 14 (hexadezimal)

B4: *-RCX*

0000

:14

A2: Das Datum wird vom aktuellen Systemdatum abgeleitet

Länge: 20 Byte

Datum: 20.12.97

F3: *PRINT TEXT1.DAT* oder

COPY TEXT1.DAT PRN

3.2 Ändern einer Datei

B1: *debug A:\TEXT1.DAT*

F1: *e105 "WAR"*

3.3 Bearbeitung von Programmdateien mit DEBUG

3.3.2 Erstellen und Testen von Programmdateien

P1: siehe Datei "ADDIERE1"

Befehl

MOV AL,79

MOV BL,AE

MOV CL,D9

ADD AL,BL

ADD AL,CL

INT 20

A1: Länge = 12 (dezimal) = 0C (hexadezimal)

B3: *ADDIERE.com*

A2: Befehl AL BL CL ZR/NZ CY/NC

00 00 00 NZ NC

MOV AL,79 79 00 00 NZ NC

MOV BL,AE 79 AE 00 NZ NC

MOV CL,D9 79 AE D9 NZ NC

ADD AL,BL 27 AE D9 NZ CY

ADD AL,CL 00 AE D9 ZR CY

INT 20 00 AE D9 ZR CY

B7: *e 300 5E C7*

P2: siehe Datei "ADDIERE2"

Befehl

MOV AL,[0300]

Kommentar

5E -> AL

AND AL,[0301]

*<AL> * <301> -> AL*

MOV [0302],AL

<AL> -> 302

INT 20

zurück nach DOS

A3: Ergebnis: *<0302> = 48*

A4: Wert 1: $ED = 11101101$
UND Wert 2: $4A = 01001010$
Ergebnis: $48 = 01001000$

4 Ein-/Ausgabefunktionen des DEBUG

4.1 Ausgabe eines Byte über eine Schnittstelle

A1: Erzeugtes Bitmuster: $F0 = 11110000$

F1: $o 378 24$

B4: Rückgabewert: 24

A2: Der Eingabewert ist immer gleich dem letzten Ausgabewert

4.2 Dateneingabe über eine Schnittstelle

B2: $C7$

A1: $C7 = 11000---$

ja

F1: Schalter H: *aus*

Taste /TE: *aus*

Schalter F: *ein*

Schalter E: *ein*

Schalter D: *ein*

4.3 Druckerstatus testen

B2: $D8$

A1: $D8 = 11011000$

Bit 2^7 : frei

Bit 2^5 : kein Papierende

Bit 2^4 : Online

Bit 2^3 : kein Fehler

B3: 40

A2: $40 = 01000000$

Bit 2^7 : belegt

Bit 2^5 : kein Papierende

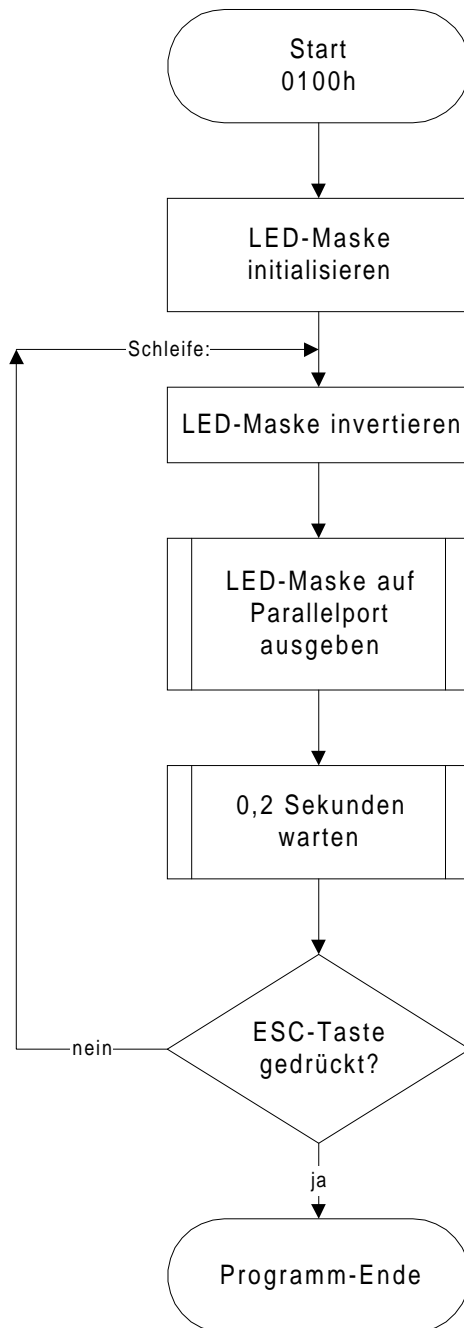
Bit 2^4 : Offline

Bit 2^3 : Fehler

B4: *Papierende erkannt (Bit $2^5 = 1$), Drucker geht in den Offline-Zustand (Bit $2^4 = 0$) und signalisiert einen Fehler (Bit $2^3 = 0$)*

4.4 Programmierübung zur Ein-/Ausgabe

P1: *Flußdiagramm*



B3: siehe Datei "BLINK"

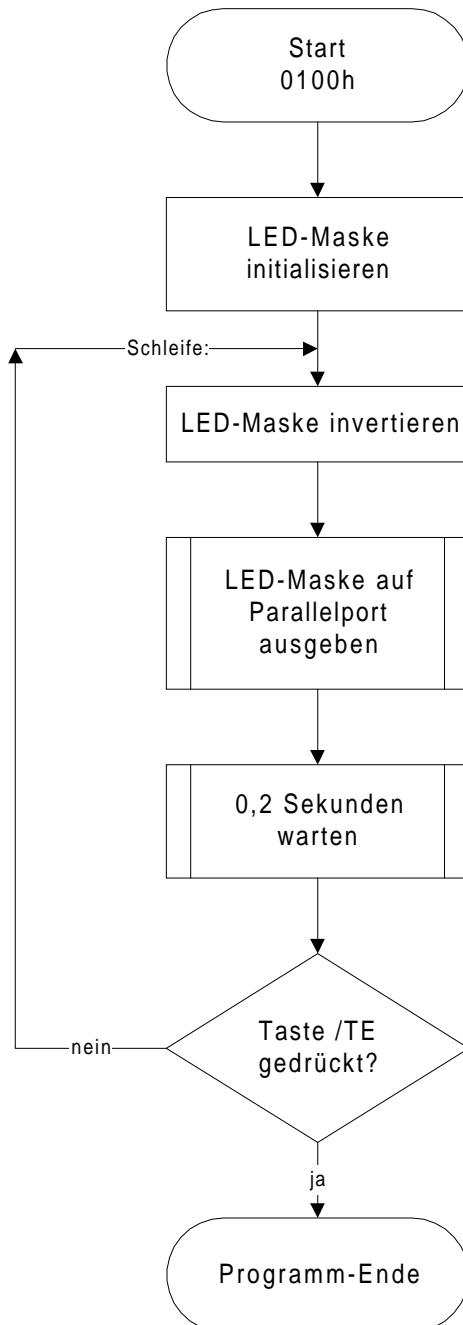
B4: cx 0000
:0380

4.5 Erweiterte Programmierübung zur Ein-/Ausgabe

F1: UND-Verknüpfung

F2: Zero-Bit

P1: Flußdiagramm



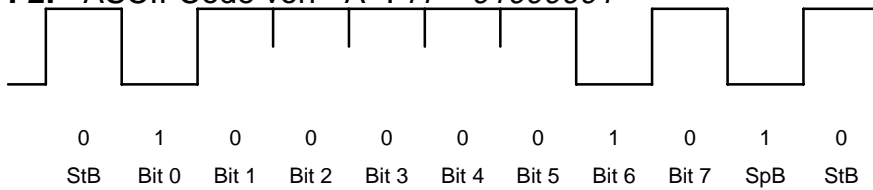
4.6 Messung an der seriellen Schnittstelle

A1: $t = \frac{1}{\text{Baudrate}} = \frac{1}{2400 \text{ Bit/s}} = 417 \mu\text{Sek.}$

M2: Spannungspegel bei logisch 0: > +3V
Spannungspegel bei logisch 1: < -3V

F1: mit dem *niedrigstwertigen* Bit

F2: ASCII-Code von »A«: $41 = 01000001$



B4: Zugehöriges MODE-Kommando: *mode com1: 96,e,8,2*

F3: 1.Befehl: Adresse 0122
2.Befehl: Adresse 0130

5 Das Interruptsystem im Personalcomputer

5.1 Interrupt-Vektortabelle

Die Werte der Interrupt-Vektoren sind systemspezifisch bedingt unterschiedlich. Sie können von der folgenden Darstellung abweichen.

M1: INT 21h		INT 60h	
Byte =	3E	Byte =	00
	10		00
	DF		00
	1F		00
Vektor =	1FDF:103E	Vektor =	0000:0000
F1:	Vektornummer * 4		
F2:	auf die ISR (Interrupt Service Routine)		
M2: INT 21h		INT 60h	
Adr. =	0000:0084	Byte =	3E
	:0085		10
	:0086		DF
	:0087		1F
Vektor =	1FDF:103E	Adr. =	0000:0180
		Byte =	A3
			CF
			56
			63
		Vektor =	6356:CFA3
F3:	Interrupt nicht belegt, dann Adresse 0000:0000, sonst Adresse <> 0000:0000		

5.2 Installieren eines speicherresidenten Programmes

siehe Datei "INT61"

A2: Adresse in der	vor	nach
Vektortabelle	Ausführung	Ausführung
0000:0184	00	03
0000:0185	00	01
0000:0186	00	58
0000:0187	00	31
A3: vor Ausführung:	0000:0000	
nach Ausführung:	3158:0103	
A4:	Sollwerte = Istwerte!	
P1: Befehl	Kommentar	
INT 61	TSR aufrufen	
INT 20	Programm beenden	
B4:	siehe Datei "INT62"	
F2:	XXXX:0144 B8 6225 MOV AX,2562	

6 Programmieren mit Software-Interrupts

6.1 Zeichenausgabe zum Bildschirm

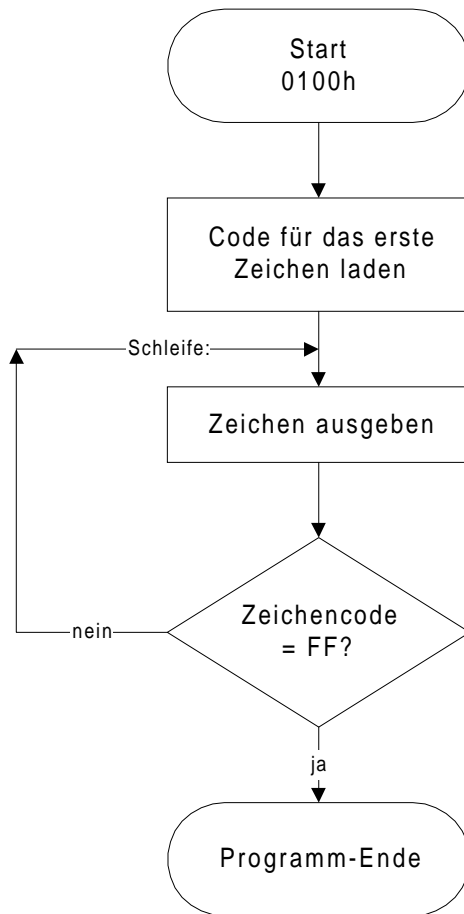
P1: siehe Datei "AUSGABE1"

P2: siehe Datei "AUSGABE2"

P3: siehe Datei "AUSGABE3"

F1: e 10B "Dies ist ein Versuch\$"

P4: Flußdiagramm



Programmliste: siehe Datei "AUSGABE4"

P5: Befehl Kommentar

`MOV AH,02h` Funktion: Zeichen zum Bildschirm

ändern in:

`MOV AH,05h` Funktion: Zeichen auf Drucker ausgeben

B8: siehe Datei "AUSGABE5"

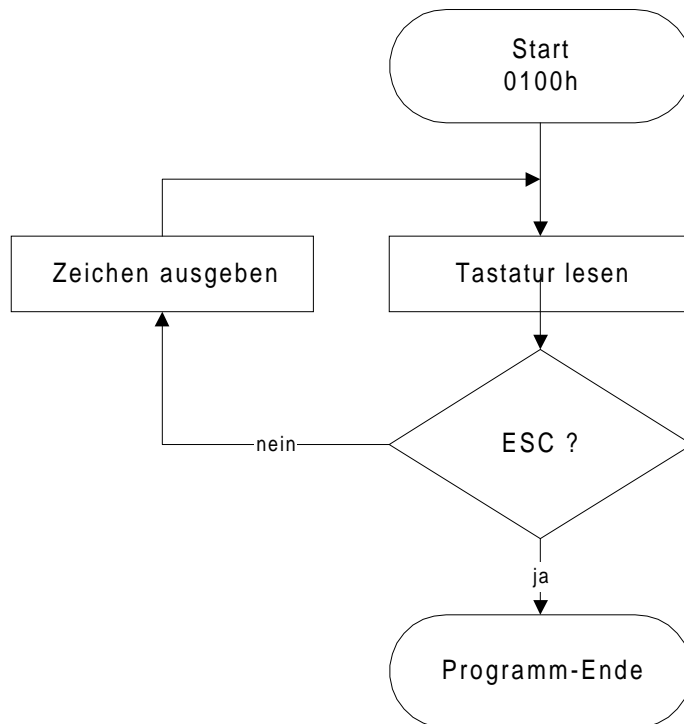
F2: Register *BL* wird mit dem Attribut geladen

Bit 7 muß auf 1 gesetzt werden, → Schrift blinkend

Befehl: `MOV BX,1E` ändern in: `MOV BX,9E`

6.2 Zeicheneingabe über die Tastatur

P1 Flußdiagramm



Programmliste: siehe Datei "ECHO1"

P2 Befehl Kommentar
 MOV AH,06h Funktion: Direkte Ein-/Ausgabe über die Konsole
 ändern in:
 MOV AH,05h Funktion: Zeichen auf Drucker ausgeben
 Programmliste: siehe Datei "ECHO2"

7 Grundübungen mit der Intelligenten Schnittstelle

7.2 Systemfunktionen der Intelligenten Schnittstelle

- P1** Programmliste: siehe Datei "HUPE"
- P2** Programmliste: siehe Datei "SLED1"
- P3** Programmliste: siehe Datei "SLED2"

7.3 Hilfsprogramme zu den Systemfunktionen der IS-535

- P1** Programmliste: siehe Datei "SYS_FUNK"
- P2** Programmliste: siehe Datei "MULTASK1"

7.4 Parallelschnittstellen der Intelligenten Schnittstelle IS-535

- F1** $11001001 = C9$
- B2** Programmliste: siehe Datei "LED_ROT"
- P2** Programmliste: siehe Datei "PAR_IO1"
- P3** Programmliste: siehe Datei "PAR_IO2"
- F2** Programmliste: *XOR BL;FF*

7.5 Hilfsprogramme zu den Parallelschnittstellen der IS-535

- P1** siehe Kapitel 7.4 (P3), Datei "PAR_IO2"

8 **Erweiterte Übungen**

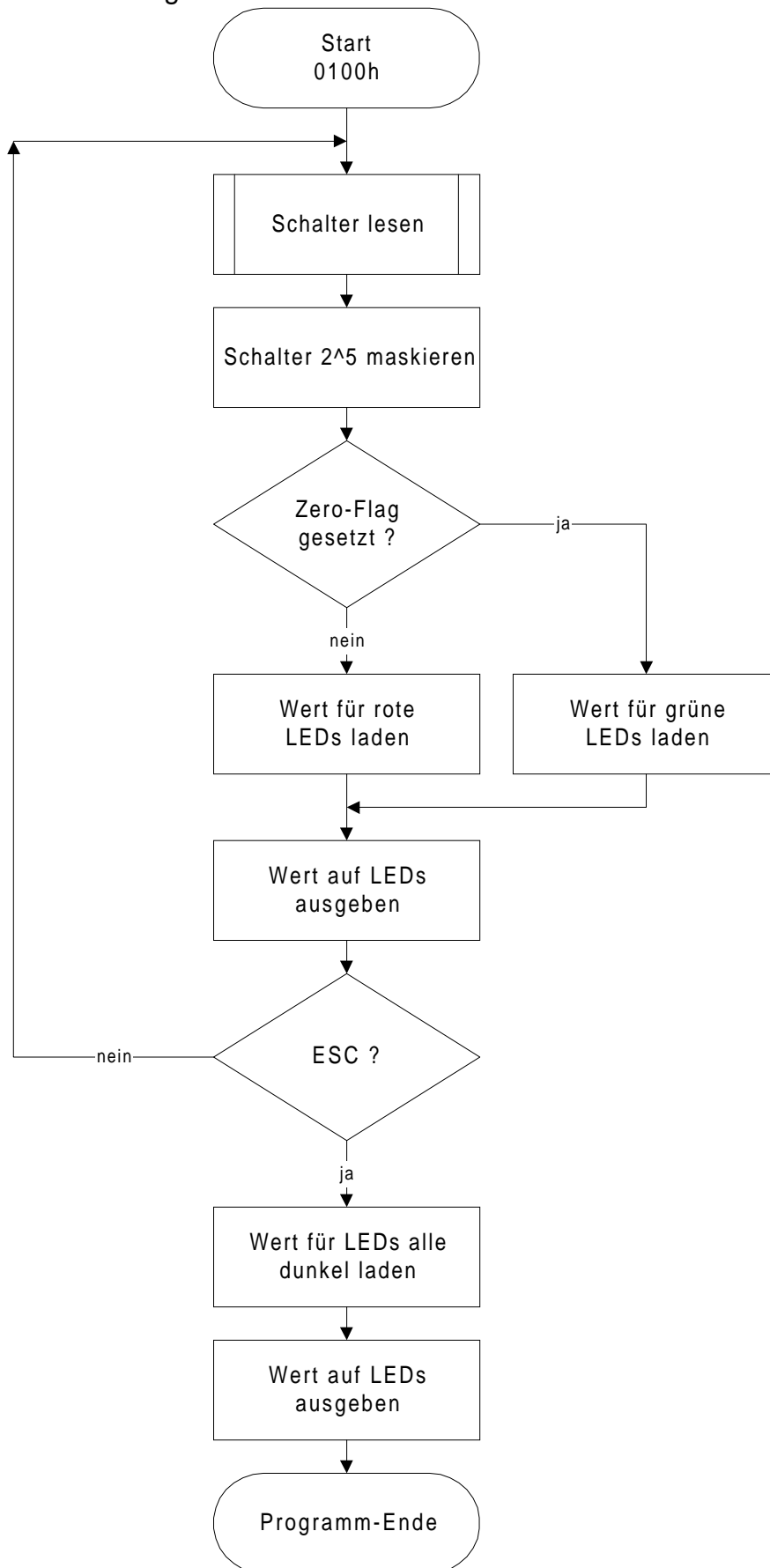
8.2 **Übungen zur paralleln Ein-/Ausgabe mit der Platine III-S9**

F1 Grüne LEDs $00100100 = 24$

 Rote LEDs $11001001 = C9$

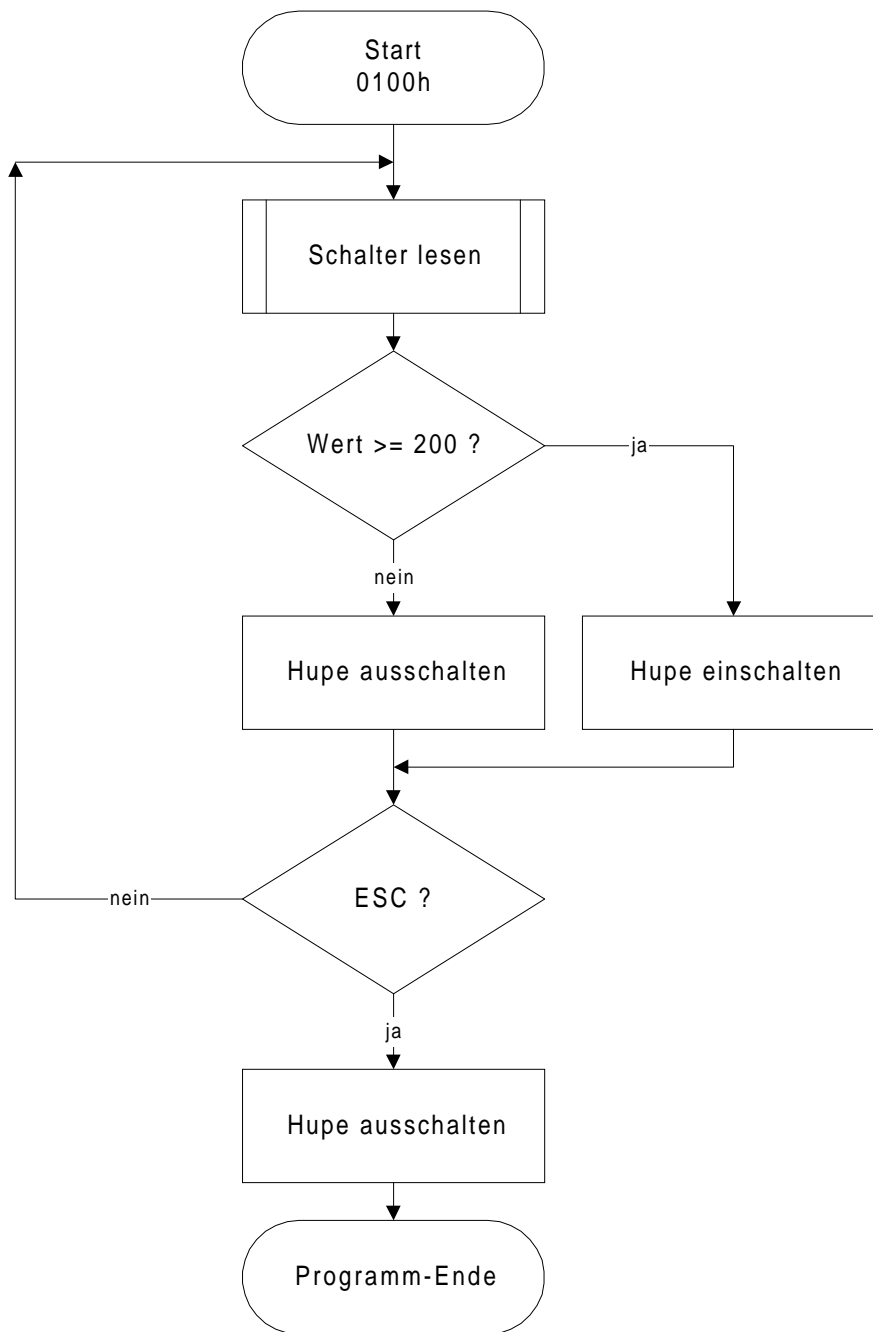
F2 Logisch-UND Verknüpfung oder TEST mit der Maske 00100000 (binär)
 danach Abfrage des Zero-Flag.

P1 Flußdiagramm



Programmliste: siehe Datei "PAR_IO3 "

P2 Flußdiagramm



Programmliste: siehe Datei "HUPE2"

8.3 Hilfsprogramm für die Platine AW-535 (Anwenderplatine)

P1 Programmliste: siehe Datei "REL_AUS"

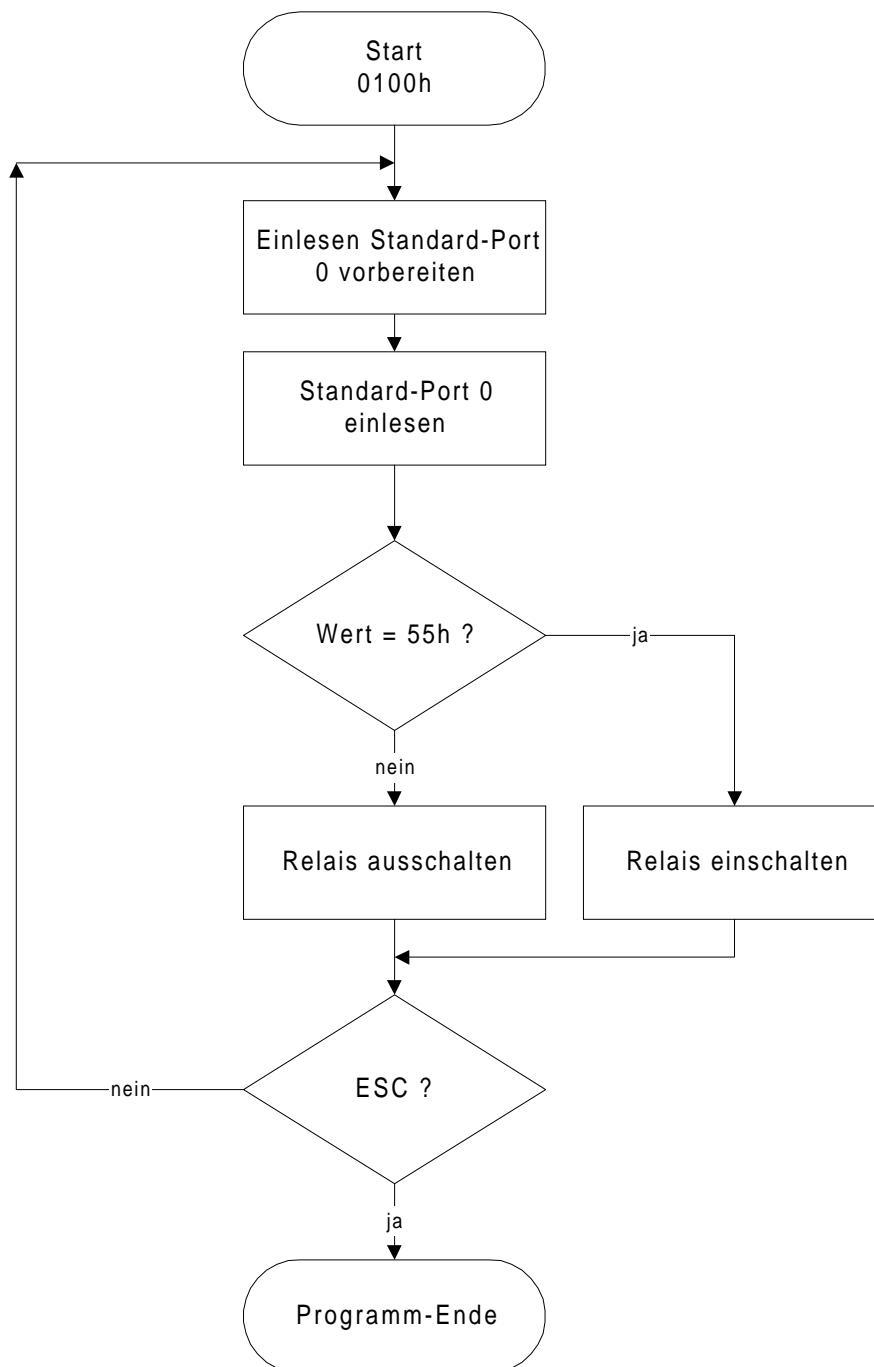
P2 anstelle Befehl `MOV AH,25h` wird Befehl `MOV AH,26h` gesetzt

P3 Programmliste: siehe Datei "BLINKER"

F1 *komplettes Abarbeiten der Zeitprogramme 1 Sek. und 3 Sek. warten*

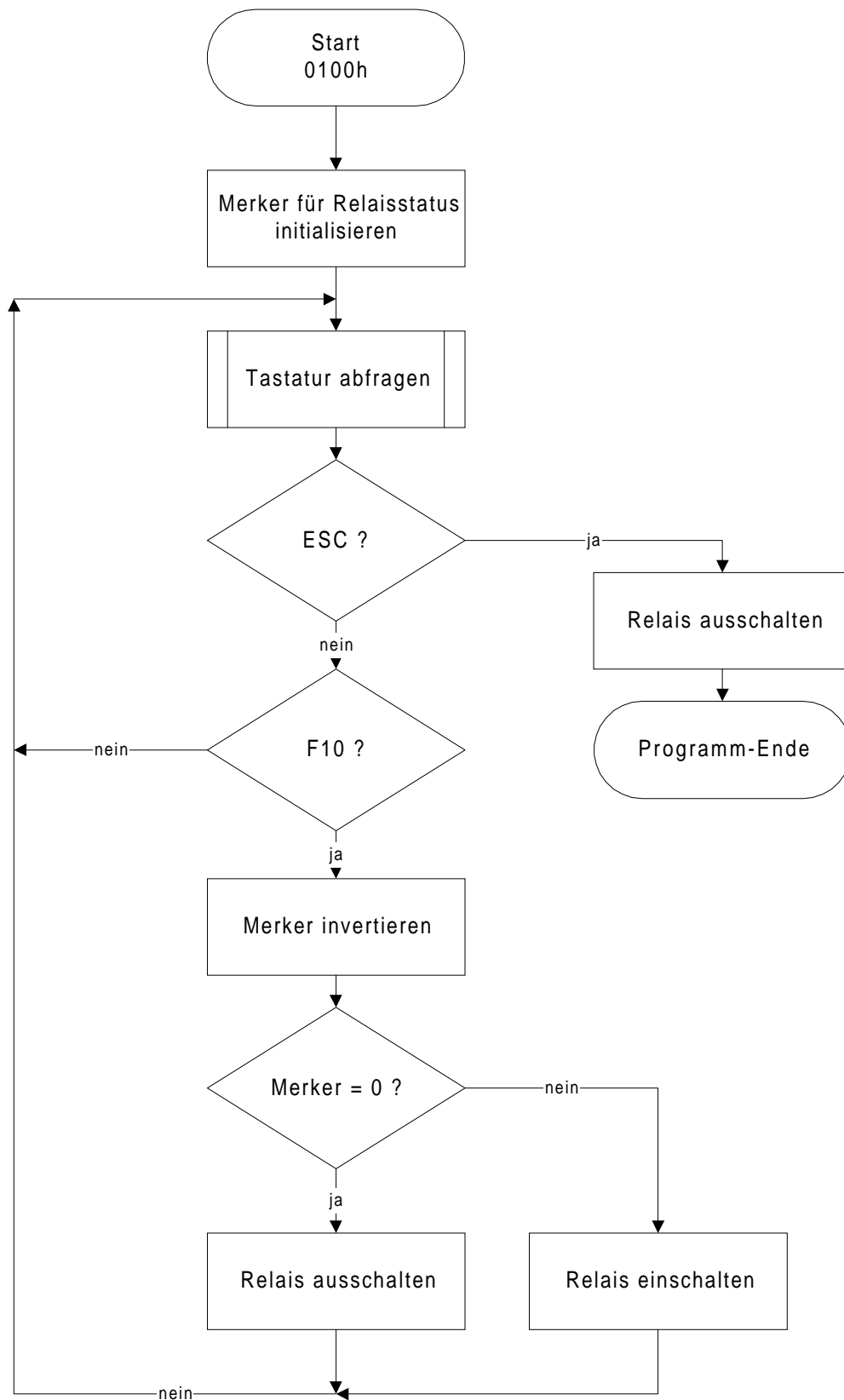
8.4 Übungen mit der Platine AW-535 (Anwenderplatine)

P1 Flußdiagramm



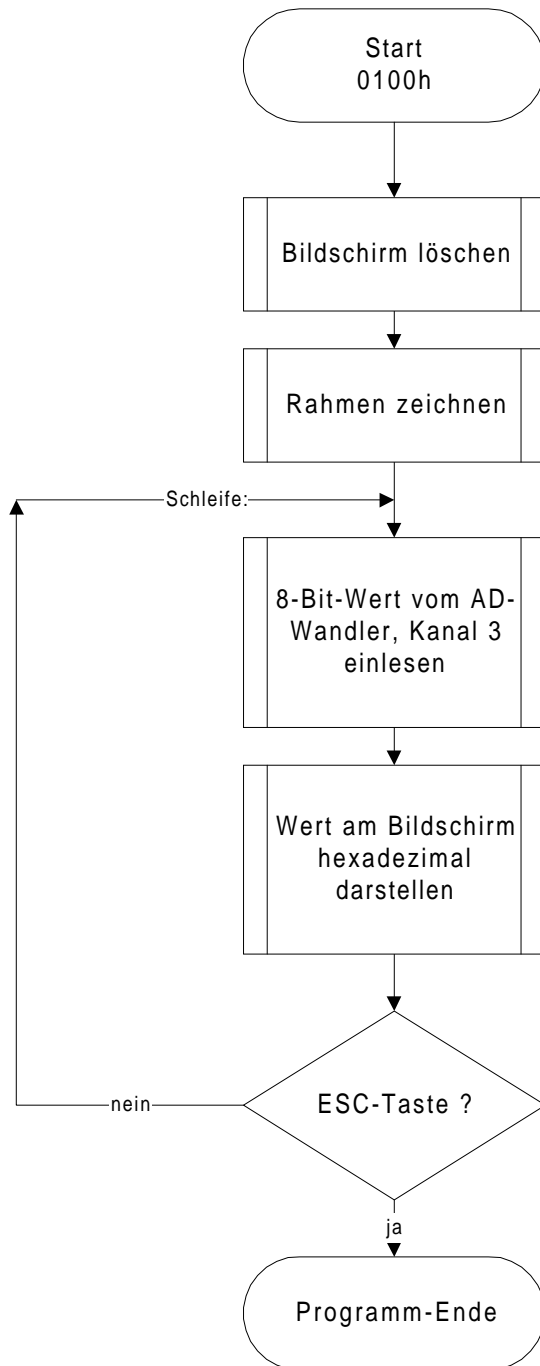
Programmliste: siehe Datei "SP0_55"

P2 Flußdiagramm



Programmliste: siehe Datei "SP0_WECH"

8.5 **Übungen zur Analog/Digital-Wandlung**
P1 Flußdiagramm



Programmliste: siehe Datei "AD_1"

F1 Schrittweite: $5V / 256 \approx 20mV$

F2 $1V / 20mV = 50 = 32h$

8.6 Analog/Digital-Wandlung

F1 Schrittweite: $\frac{U_{Ref}}{2^n} = \frac{2,56V}{2^8} = \frac{2,56V}{256} = 10mV$

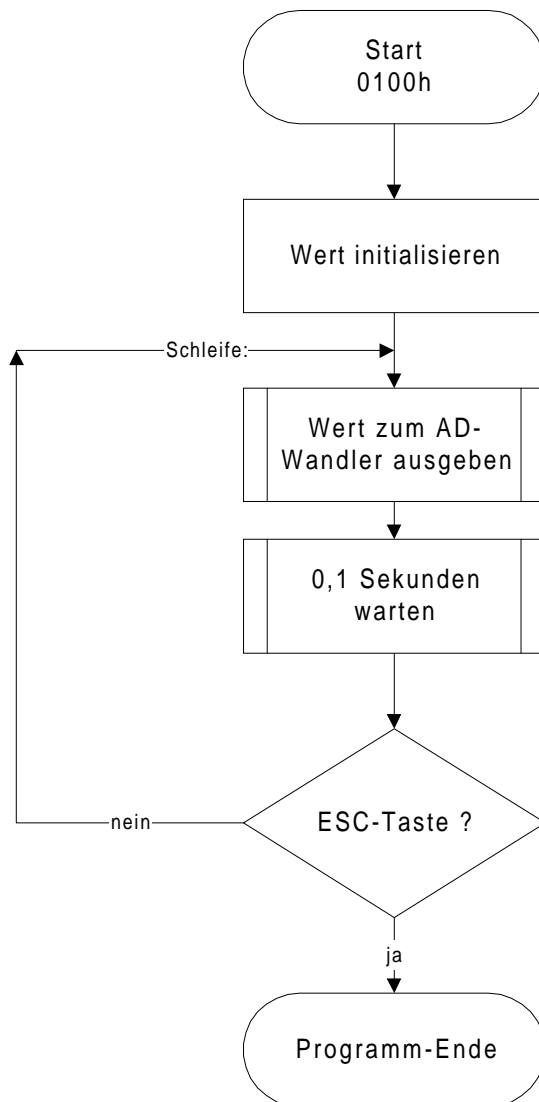
$U_{Amin} = 0V$ $U_{Amax} = 2,55V$

Programmliste: siehe Datei "AD_2"

M1 Schalterreihe SW3 einstellen und U_A messen:

Binärwert	Hex	Dez	U_A (berechnet)	U_A (gemessen)
0 0 1 1 0 0 1 0	32	50	0,50 V	
0 1 1 1 1 0 0 0	78	120	1,20 V	
1 0 1 1 1 0 0 1	B9	185	1,85 V	
1 1 0 1 1 1 1 0	DE	222	2,22 V	

P2 Flußdiagramm



Programmliste: siehe Datei "SAEGEZAH"

9 Anwendungsbeispiele

9.2 Projekt »Spannungsmessung«

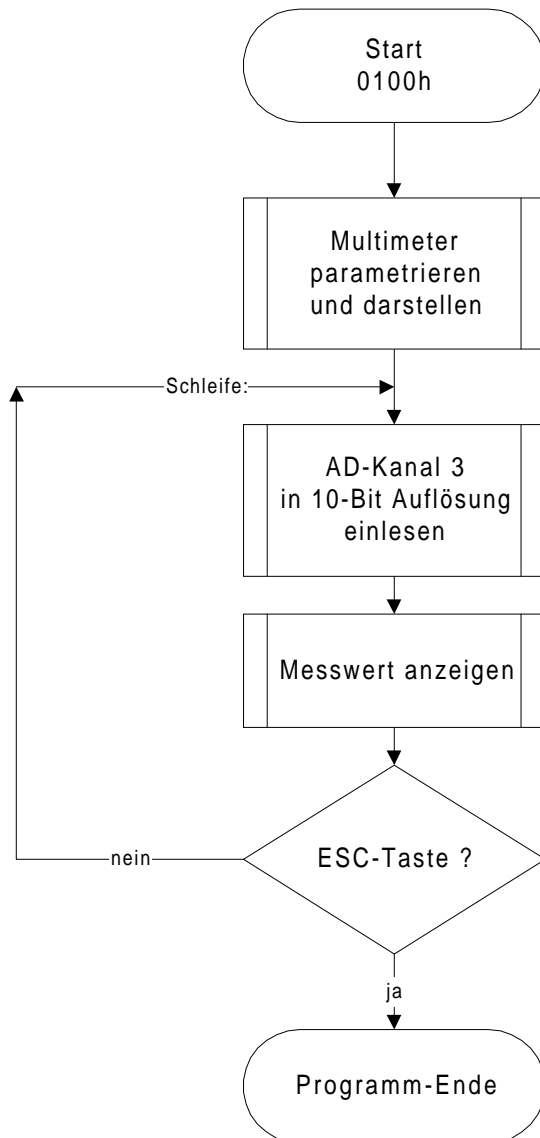
9.2.3 Aufgabenstellung

$$\text{Schrittweite} = \frac{U_{\text{Ref}}}{2^{10}} = \frac{5 \text{ V}}{1024} = 4,88 \text{ mV}$$

9.2.4 Parametrierung des Programms MULTIMET.BIN

Die Parametrierung des Multimeters erfolgt im Programm SPG_MESS im Unterprogramm INITMULTIMETER. Sie kann auch menügeführt durch das Programm MUSE.EXE erfolgen.

9.2.5 Flußdiagramm

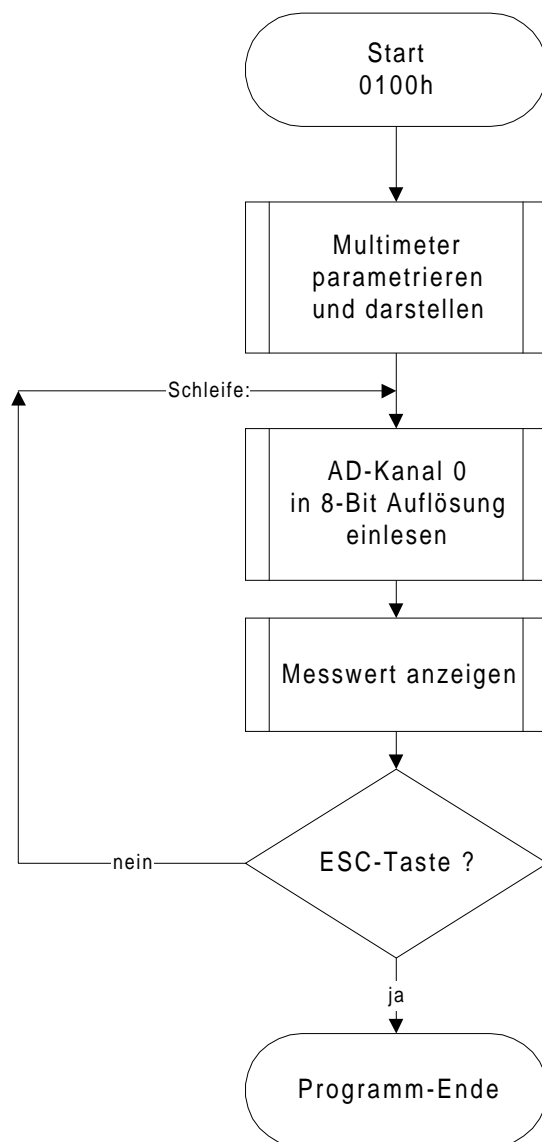


9.2.6 Programmliste
Programmliste: siehe Datei "SPG_MESS"

9.3 Projekt »Temperaturmessung«

9.3.4 Parametrierung des Programms MULTIMET.BIN
Die Parametrierung des Multimeters erfolgt im Programm TMP_MESS im Unterprogramm INITMULTIMETER. Sie kann auch menügeführt durch das Programm MUSE.EXE erfolgen.

9.3.5 Flußdiagramm

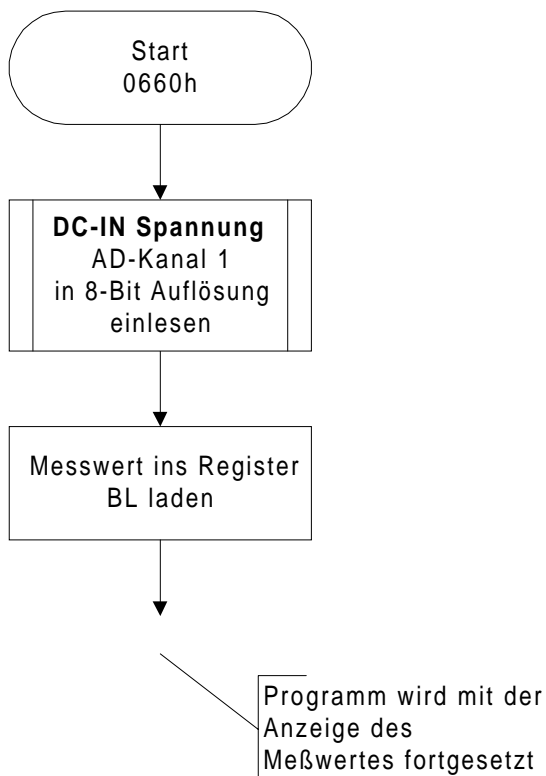
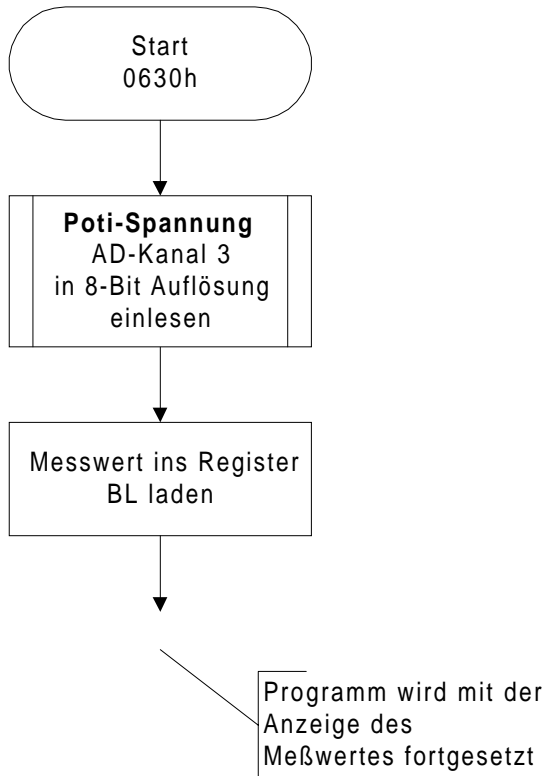


9.3.6 Programmliste

Programmliste: siehe Datei "TMP_MESS"

9.4 Projekt »Meßstellenüberwachung«

9.4.4 Flußdiagramm



9.4.5 Programmliste

Programmliste: siehe Datei "MS_UE_WA"

9.5 Projekt »Motorsteuerung«

9.5.3 Manuelle Drehzahlsteuerung

Niedrigste Drehzahl: 1180

Höchste Drehzahl: 4840

9.5.5 Drehzahl-Parameter vom PC aus einstellen

Niedrigste Drehzahl: 1750

Höchste Drehzahl: 3960

9.5.7 Drehzahl-Parameter vom PC aus einstellen

Tachometer-Kennwert: $1V / 2000 \text{ min}^{-1}$

9.6 Projekt »Zugangskontrolle«

9.6.5 Programmliste

Programmliste: siehe Datei "ZUG_KONT"

Auszug aus der Programmliste:

```

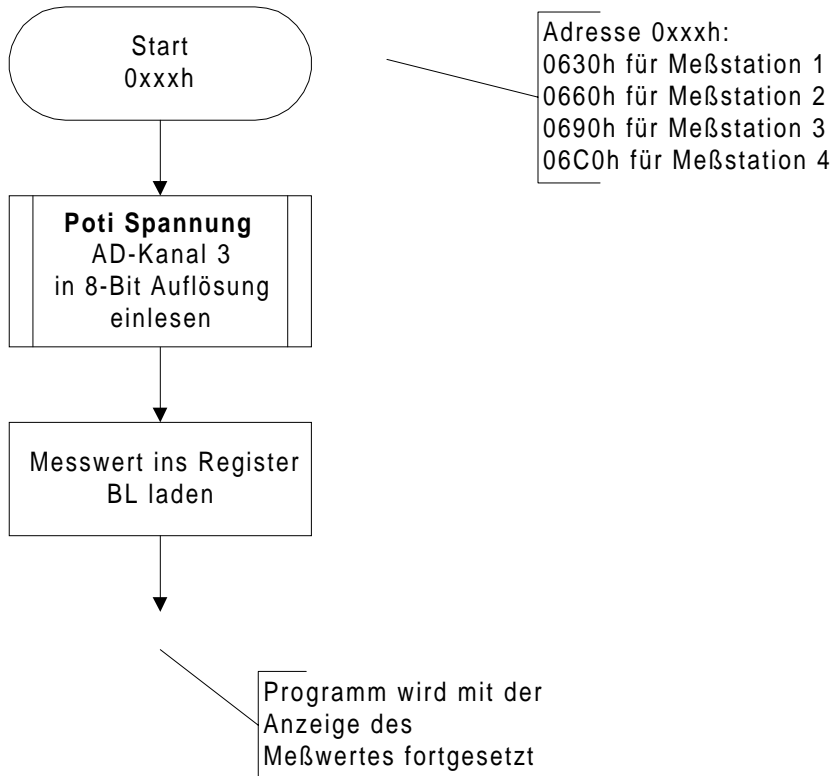
0000          ;
0000          ;  Hauptprogramm
0000          ;
0100          ORG      0100h
0100          Start:
0100 E8 4D01   CALL    GLED_EIN      ; Grüne LED einschalten
0103 E8 AA01   CALL    RELAIS_AUS    ; Relais ausschalten
0106 E8 5701   CALL    RLED_AUS     ; Rote LED ausschalten
0109 E8 1401   CALL    HUPE_AUS     ; Signalhupe ausschalten
010C          CALL    ERFASSEN     ; Eingabe der Codezahl
010F          CALL    GLED_AUS     ; Grüne LED ausschalten
0112 E8 2503   CALL    ZEIT10      ; Zeitprogramm 1,0 Sekunden

0115 E8 5800   CALL    VERGLEICH    ; Vergleich der Codezahl mit der
0118          ; Geheimzahl
0118 80 FE 00  CMP    DH,00h      ; und Verzweigung
011B 74 09     JZ     gleich
011D          ungleich:
011D E8 5001   CALL    RLED_EIN     ; Rote LED der IS-535 einschalten
0120 E8 0D01   CALL    HUPE_EIN     ; Signalhupe der IS-535 einschalten
0123 E9 1A00   JMP    ENDABFRAGE    ; zur Tastenabfrage
0126          gleich:
0126 E8 A701   CALL    HUPE_05      ; Signalhupe der IS-535 für
0129          ; 0,5 Sek. einschalten
0129 E8 9401   CALL    RELAIS_EIN   ; Relais auf AW-535 einschalten
012C E9 1100   JMP    ENDABFRAGE    ; zur Tastenabfrage

```

9.7 Projekt »Vernetzte Meßstationen«

9.7.3 Flußdiagramm



9.7.5 Programmliste

Programmliste: siehe Datei "MES_NETZ"

Auszug aus der Programmliste:

```

;
; Routine für Meßstation 1 - Poti an Station 1
;
ORG      0630h
0630
0630 B4 41      MOV      AH,AD_8_EINZ      ; Funktion: einen 8-Bit-Wert lesen
0632 B0 01      MOV      AL,Station1
0634 B7 03      MOV      BH,Kanal3        ; Nummer des AD-Kanals
0636 CD 60      INT      HPI          ; Wert nach DX lesen
0638 88 D3      MOV      BL,DL          ; Messwert nach BL
063A
063A
;
; Routine für Meßstation 2 - Poti an Station 2
;
ORG      0660h
0660
0660 B4 41      MOV      AH,AD_8_EINZ      ; Funktion: einen 8-Bit-Wert lesen
0662 B0 02      MOV      AL,Station2
0664 B7 03      MOV      BH,Kanal3        ; Nummer des AD-Kanals
0666 CD 60      INT      HPI          ; Wert nach DX lesen
0668 88 D3      MOV      BL,DL          ; Messwert nach BL
066A
066A
;
; Routine für Meßstation 3 - Poti an Station 3
;
ORG      0690h
0690
0690 B4 41      MOV      AH,AD_8_EINZ      ; Funktion: einen 8-Bit-Wert lesen
0692 B0 03      MOV      AL,Station3
0694 B7 03      MOV      BH,Kanal3        ; Nummer des AD-Kanals
0696 CD 60      INT      HPI          ; Wert nach DX lesen
0698 88 D3      MOV      BL,DL          ; Messwert nach BL
069A
069A
;
; Routine für Meßstation 4 - Poti an Station 4
;
ORG      06C0h
06C0
06C0 B4 41      MOV      AH,AD_8_EINZ      ; Funktion: einen 8-Bit-Wert lesen
06C2 B0 04      MOV      AL,Station4
06C4 B7 03      MOV      BH,Kanal3        ; Nummer des AD-Kanals
06C6 CD 60      INT      HPI          ; Wert nach DX lesen
06C8 88 D3      MOV      BL,DL          ; Messwert nach BL

```