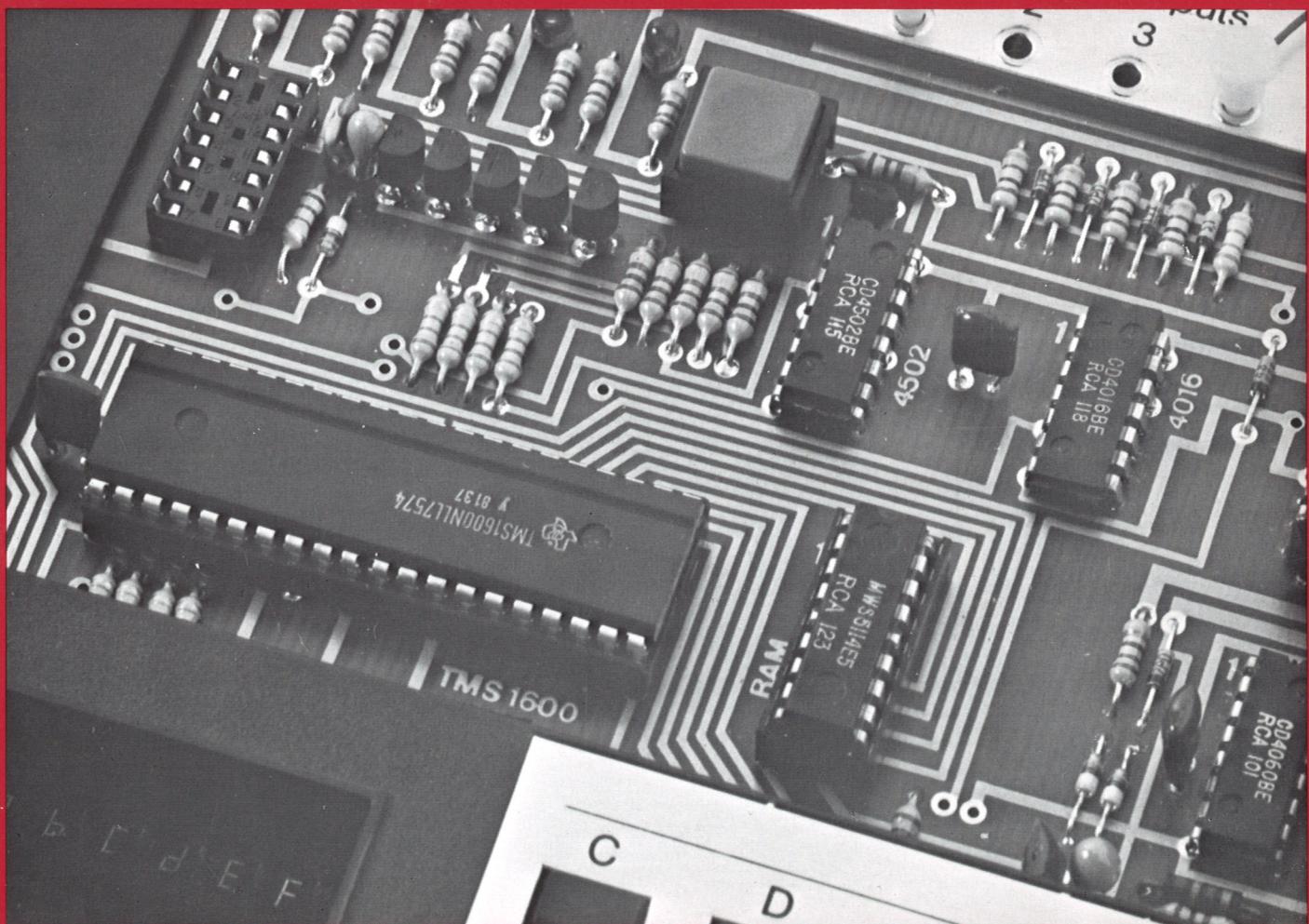


**microtronic** computer system

**2090**

**Anleitungsbuch  
2. Teil**

**Programmieren – Experimentieren – spielend lernen  
wie ein Computer funktioniert.**



In Zusammenarbeit mit  
dem Elektronik-Magazin



## Druckfehlerberichtigung im Microtronic-Anleitungsbuch 2. Teil

Auf Seite 44: Schaltuhr mit zwei Ein- und Ausschalt-Zeiten.

Der Eingabe-Befehl bei Adresse 57 muß richtig heißen: 529 (anstelle D29).

Seite 62: Computer zählt Frequenzen

Das im Buch ausgedruckte Programm endet bei Adresse 1B. Es muß wie folgt ergänzt werden:

1C	0A8	MOV A,8
1D	0B9	MOV B,9
1E	080	MOV 8,0
1F	091	MOV 9,1
20	C01	GOTO 01



Bitte entschuldigen Sie das Versehen.



## Anleitungsbuch 2. Teil

Einführung in die  
Mikroprozessor-Technik.

Programme zum  
Experimentieren  
mit dem Microcomputer.

Von Jörg Vallen



Batterie  
9V

Bei einigen Experimenten  
zusätzlich erforderlich:  
Eine Batterie 9 V (IEC 6 F 22)  
oder Netzteil 2059

Produktion und Vertrieb:  
**BUSCH-Modellspielwaren**  
**Postfach 1360**  
**D 6806 Viernheim**

In Zusammenarbeit mit dem  
**ELO-Magazin Franzis-Verlag**  
**Postfach 370120**  
**8000 München 37**

Copyright 1981 by  
BUSCH GmbH. + Co. KG, Vierneheim  
Alle Rechte vorbehalten.  
Grafiken  
Atelier Wuthe, Weinheim  
Printed in W.-Germany  
10/81

In diesem Teil des Anleitungsbuches wird aus der großen Palette der Anwendungsmöglichkeiten des Microtronic-Computer-systems eine kleine Programm-Auswahl demonstriert. Bei den Programm Erklärungen wird davon ausgegangen, daß der 1. Teil des Anleitungsbuches bereits durchgearbeitet wurde.

Durch die einleitenden Erklärungen ist es jedoch auch ohne detaillierte Computer-Kenntnisse möglich, die aufgeführten Programme einzugeben, um interessante Computer-Spiele und andere Möglichkeiten kennenzulernen.

## Folgende Programme stehen zur Auswahl:

<b>Computer-Spiele</b>	Seite
Tic-Tac-Toe	4
See-Schlacht	6
Code-Brecher (Denk- und Geschicklichkeits-Spiel)	9
Nimm 2	12

## Computer Funktionen mit Decoder-Aufgaben

Morsezeichen-Decoder	18
Morse-Trainer	21
Widerstands-Code-Berechnung	24

## Computer-Berechnungen

Taschenrechner-Programm (mit sämtlichen Rechenoperationen)	26
Sinus-Berechnung (etwas für Mathematiker)	32
Berechnung der Tage zwischen zwei Daten	34
Aus eingegebenen Daten einen Wochentag berechnen	36
Berechnung des persönlichen Bio-Rhythmus	37

## Computer-Funktionen mit Peripherie-Elektronik

Für diese Experimente sind zusätzliche BUSCH-Schaltrelais, bzw. BUSCH-Electronic-Studios erforderlich.

Anschluß von Relais	42
Schaltuhr mit zwei Ein- und Ausschalt-Zeiten	44
Weck-Uhr mit zwei Weckzeiten	47
Wecker mit Schlummer-Radio	48
Weitere „Schaltuhr“-Möglichkeiten	48
Computer-Orgel	52
Der komponierende Computer	54
Computer steuert elektronische Spieluhr	54
Akustische Effekte für Computer-Spiele	56
Der Computer zählt Personen und Gegenstände	57
Daten-Übertragung mit nur 3 Leitungen	59
Daten-Übertragung im „hand-shaking“-Verfahren	61
Der Computer zählt Frequenzen	62
Reaktionszeit-Meßgerät	65
Digital-Voltmeter	68
Computergesteuerte Modelleisenbahn	71

Diese Programme dienen als Anregung wie der Computer für die verschiedensten Aufgaben immer wieder neu programmiert werden kann.

## Microtronic Computer-Anfänger bitte beachten:

Vor der Programmierung sind jeweils die Tasten HALT – NEXT – 00 zu betätigen. Dann die Befehls-Codes gemäß Programm-Tabelle eingeben. Nach jeder Befehls-Code Eingabe nicht vergessen, die NEXT-Taste zu betätigen. Nach Eingabe des letzten Befehls-Codes eines Programms nochmals die Taste NEXT, dann mit HALT – NEXT – 00 – RUN das Programm starten. Bei nicht einwandfreier Programm-Funktion nochmals mit HALT – NEXT – 00 zum Programm-Anfang gehen und durch Betätigung der NEXT-Taste die eingegebenen Befehle mit der Programm-Tabelle vergleichen und notfalls korrigieren (neu eingeben). Nach durchgeföhrter Korrektur nicht vergessen die NEXT-Taste zu betätigen und dann mit HALT – NEXT – 00 – RUN das Programm erneut starten.

## Tic-Tac-Toe

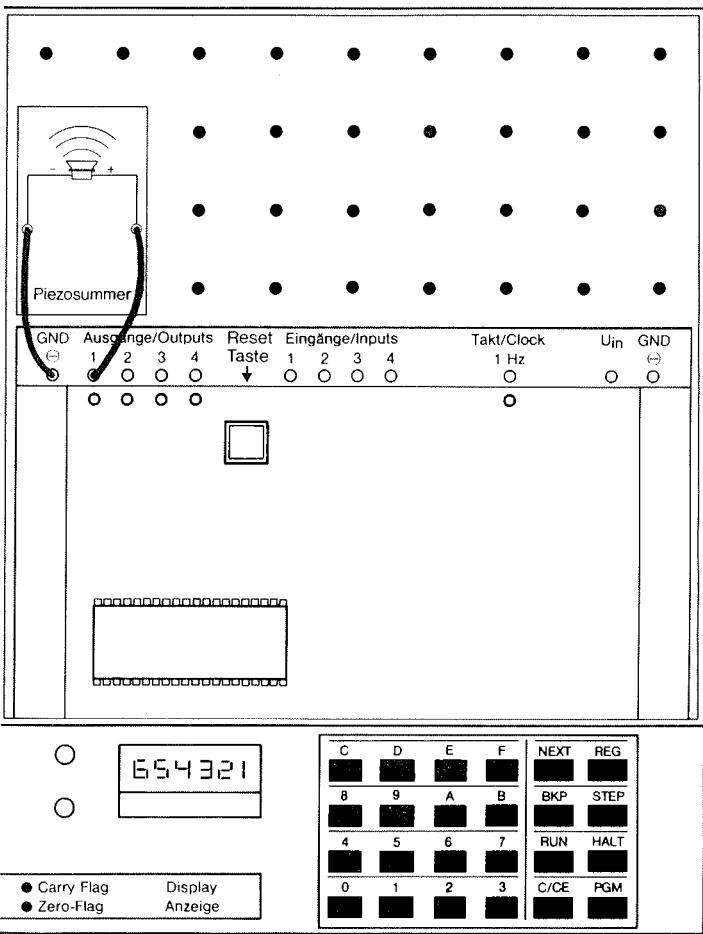
## **Programm: Tic – Tac – Toc**

Tic-Tac-Toe ist ein altbekanntes, einfaches Brett-Spiel. Es hat ein Spielfeld mit 9 Felder, wobei  $3 \times 3$  Felder quadratisch angeordnet sind (siehe Abbildung). Jeder Spieler hat 4 Steine. Es wird abwechselnd gespielt, wobei jeder Spieler einen Stein auf ein freies Feld absetzen kann. Es gewinnt der Spieler, welcher zuerst 3 Steine in einer waagrechten, senkrechten oder diagonalen Reihe anordnen kann.

Beim Computer-Zubehör finden wir ein Tic-Tac-Toe-Spielfeld mit jeweils 4 schwarzen und roten Spielmarken.

Zunächst müssen wir nach HALT – NEXT – 00 das Programm lt. Tabelle eingeben.

### **Anschluß des Piezosummers**



Sprungziel	Adresse	Eingabe-Befehl	Mnemonic	Sprung nach
	00	<b>F08</b>	CLEAR	
	01	<b>F0D</b>	EXRL	
	02	<b>F08</b>	CLEAR	
	03	<b>FE0</b>	DOT 0	
	04	<b>198</b>	MOVI 9,8	
	05	<b>F28</b>	DISP 2,8	
	06	<b>FF9</b>	KIN 9	
	07	<b>51A</b>	ADDI 1,A	
	08	<b>94A</b>	CMPI 4,A	
	09	<b>E21</b>	BRZ 21	„EE“
	0A	<b>09D</b>	MOV 9,D	
	0B	<b>51D</b>	ADDI 1,D	
	0C	<b>B2B</b>	CALL 2B	„BERE“
	0D	<b>0C8</b>	MOV C,8	
	0E	<b>FF9</b>	KIN 9	
	0F	<b>08D</b>	MOV 8,D	
	10	<b>54D</b>	ADDI 4,D	
	11	<b>B2B</b>	CALL 2B	„BERE“
	12	<b>89C</b>	CMP 9,C	
	13	<b>E07</b>	BRZ 07	
	14	<b>0C8</b>	MOV C,8	
	15	<b>1CA</b>	MOVI C,A	
	16	<b>1CB</b>	MOVI C,B	
	17	<b>F48</b>	DISP 4,8	
	18	<b>10F</b>	MOVI 0,F	
	19	<b>FEF</b>	DOT F	
	1A	<b>516</b>	ADDI 1,6	
	1B	<b>D1F</b>	BRC 1F	
	1C	<b>1FF</b>	MOVI F,F	
	1D	<b>FEF</b>	DOT F	
	1E	<b>C18</b>	GOTO 18	
	1F	<b>FF0</b>	KIN 0	
	20	<b>C00</b>	GOTO 00	
„EE“	21	<b>09D</b>	MOV 9,D	
	22	<b>55D</b>	ADDI 5,D	
	23	<b>02B</b>	CALL 2B	„BERE“
	24	<b>0C8</b>	MOV C,8	
	25	<b>1EA</b>	MOVI E,A	
	26	<b>1EB</b>	MOVI E,B	
	27	<b>1FF</b>	MOVI F,F	
	28	<b>FEF</b>	DOT F	
	29	<b>F48</b>	DISP 4,8	
	2A	<b>C1F</b>	GOTO 1F	
„BERE“	2B	<b>0DC</b>	MOV D,C	
	2C	<b>71D</b>	SUBI 1,D	
	2D	<b>10E</b>	MOVI 0,E	
	2E	<b>F03</b>	HDXZ	
	2F	<b>0D0</b>	MOV D,0	
	30	<b>0E1</b>	MOVI E,1	
	31	<b>F0D</b>	EXRL	
	32	<b>180</b>	MOVI 8,0	
	33	<b>F0C</b>	DIV	
	34	<b>F0D</b>	EXRL	
	35	<b>101</b>	MOVI 0,1	
	36	<b>180</b>	MOVI 8,0	
	37	<b>F0B</b>	MULT	
	38	<b>60C</b>	SUB 0,C	
	39	<b>F07</b>	RET	

Jetzt den Piezo-Summer (siehe Abb. unten) anschließen. Dann Programm-Start: HALT – NEXT – 00 – RUN

Das Display zeigt: **09**. Dies bedeutet, daß der Computer seinen ersten Spielstein auf Feld 9 setzt. Wir helfen ihm, indem wir eine rote Spielmarke auf Feld 9 ablegen.

Wir setzen einen weißen Stein auf Feld 3 und geben die Ziffer **3** ein. Das Display zeigt jetzt: **34**. Hiermit bestätigt der Computer, daß er unser Feld 3 registriert hat, und daß er selbst seinen roten Stein auf Feld **4** wünscht.

Damit der Computer die quer verlaufende Spielstein-Reihe nicht auffüllen kann, setzen wir einen weißen Stein auf Ziffer 8 (**8** eingeben). Das Display meldet: **81** – Der Computer möchte einen roten Stein auf Feld **1** haben.

Wir setzen z. B. unseren weißen Stein auf Feld 6 (**6** eingeben). Das Display meldet: CC65 und es ertönt ein unterbrochener Summtton.

Die Meldung **CC65** besagt, daß der Computer seinen nächsten roten Stein auf das Feld 5 absetzen wird, wodurch jetzt 3 rote Steine in einer diagonalen Reihe angeordnet sind. Der Computer hat gewonnen.

Sobald eine beliebige Zahlen-Taste betätigt wird, kann ein neues Spiel begonnen werden, wobei der Computer seinen ersten Stein wieder auf dem Feld 9 absetzt.

Sollte es uns gelingen, ein „Unentschieden“ zu erkämpfen, meldet das Display: **EE** und zweimal die Ziffer, die wir als letztes eingegeben haben. Dies wäre auch die Ziffer gewesen, welche der Computer zu einem Sieg benötigt hätte. Nachdem keiner der Spieler eine fortlaufende Reihe erreichen kann, wird das Spiel mit „Unentschieden“ abgebrochen. Es ertönt ein Dauerton solange, bis durch die Betätigung einer Zahlen-Taste ein neues Spiel beginnt.

Wir werden feststellen, daß es unmöglich ist, gegen den Computer zu gewinnen. Ein „Unentschieden“ ist die beste Möglichkeit, die von uns erreicht werden kann.

#### **Das interessiert den Programmierer:**

Die eigentlichen Berechnungen des Computers werden in einem Unterprogramm vorgenommen (von Adresse 2B bis Adresse 39). Diese Subroutine „BERE“ (BERECHNUNG) ist auch im Programm-Ablaufplan dargestellt. Die Subroutine „BERE“ wird von den Adressen 0C oder 11 oder 23 angesprungen.

Zunächst wird der Wert des Registers D ins Register C gespeichert. Danach berechnet der Computer folgende Formel:

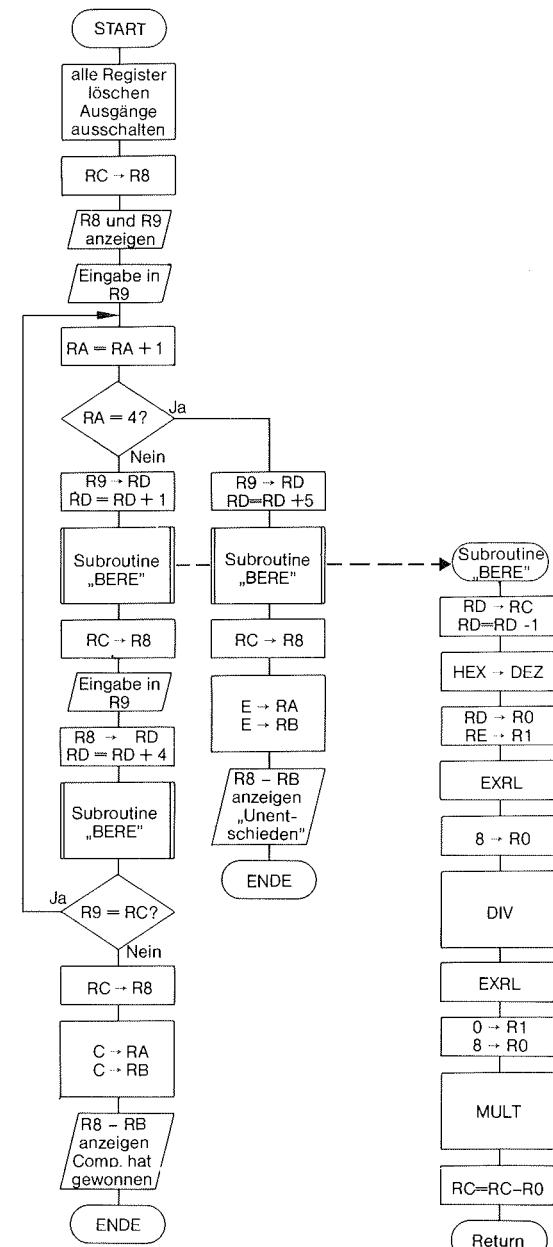
$$\text{RC (Register C)} = \text{RC} - 8 \times ((\text{RD} - 1) : 8)$$

Der Wert in Register D wird um den Wert 1 verkleinert und durch 8 dividiert. Angenommen, in Register D steht der Wert 4, wird dieser Wert um 1 vermindert, verbleiben  $3 \cdot 3 \div 8$  ergibt 0 Rest 3. Der Restwert wird bei den folgenden Berechnungen nicht benötigt, weshalb die folgende Berechnung:  $8 \times 0 = 0$  ergibt. Nachdem im Register C der Wert 4 vorhanden war, der Wert 0 abgezogen wird, bleiben 4 übrig. Durch diese intelligente Formel wird sichergestellt, daß immer ein Feld zwischen 1 und 9 möglichst in einer Reihe oder in einer Diagonalen besetzt wird.

Der Vergleich Programm-Tabelle mit Programm-Ablaufplan zeigt uns die einzelnen Funktionen.

Bei der Programm-Tabelle wurden in der Spalte „Sprung-Ziel“ kenntlich gemacht, bei welchen Adressen die angesprochenen Programm-Teile, bzw. Unter-Programme (Subroutine) beginnen. In der Spalte „Sprung nach“ wurde ebenfalls kenntlich gemacht, bei welchen Adressen die Sprünge beginnen.

#### **Programm-Ablaufplan = Tic-Tac-Toe**



## „See-Schlacht“ oder „Kampf im Weltraum“

Diesem Spiel könnte man viele Namen geben. Die Strategie ist folgende: Auf einem Spielfeld mit  $5 \times 5 = 25$  Felder versteckt der Computer 4 feindliche Schiffe. Durch Eingabe der Feld-Koordinaten müssen die Schiffe gefunden und abgeschossen, bzw. versenkt werden.

Zunächst ist das Spiel zu programmieren. Nach HALT – NEXT – 00 nehmen wir sorgfältig die Eingabe vor (Piezo-Summer abklemmen):

### Programm: See-Schlacht

Sprungziel	Adresse	Eingabe-Befehl	Mnemonic	Sprung nach
	00	<b>F02</b>	DISOUT	
	01	<b>F08</b>	CLEAR	
	02	<b>FE0</b>	DOT 0	
	03	<b>B53</b>	CALL 53	RND
	04	<b>0D0</b>	MOV D,0	
	05	<b>0E1</b>	MOV E,1	
	06	<b>B53</b>	CALL 53	RND
	07	<b>B6D</b>	CALL 6D	V1
	08	<b>D06</b>	BRC 06	
	09	<b>0D2</b>	MOV D,2	
	0A	<b>0E3</b>	MOV E,3	
	0B	<b>B53</b>	CALL 53	RND
	0C	<b>B68</b>	CALL 68	V2
	0D	<b>D0B</b>	BRC 0B	
	0E	<b>0D4</b>	MOV D,4	
	0F	<b>0E5</b>	MOV E,5	
	10	<b>B53</b>	CALL 53	RND
	11	<b>B63</b>	CALL 63	V3
	12	<b>D10</b>	BRC 10	
	13	<b>0D6</b>	MOV D,6	
	14	<b>0E7</b>	MOV E,7	
	15	<b>10D</b>	MOVI 0,D	
	16	<b>10E</b>	MOVI 0,E	
	17	<b>10F</b>	MOVI 0,F	
	18	<b>F28</b>	DISP 2,8	
	19	<b>FF9</b>	KIN 9	
	1A	<b>F28</b>	DISP 2,8	
	1B	<b>FF8</b>	KIN 8	
	1C	<b>F02</b>	DISOUT	
	1D	<b>51D</b>	ADDI 1,D	
	1E	<b>FBE</b>	ADC E	
	1F	<b>880</b>	CMP 8,0	
	20	<b>E22</b>	BRZ 22	
	21	<b>C24</b>	GOTO 24	
	22	<b>891</b>	CMP 9,1	
	23	<b>E38</b>	BRZ 38	R0
	24	<b>882</b>	CMP 8,2	
	25	<b>E27</b>	BRZ 27	
	26	<b>C29</b>	GOTO 29	
	27	<b>893</b>	CMP 9,3	
	28	<b>E3A</b>	BRZ 3A	
	29	<b>884</b>	CMP 8,4	
	2A	<b>E2C</b>	BRZ 2C	
	2B	<b>C2E</b>	GOTO 2E	
	2C	<b>895</b>	CMP 9,5	

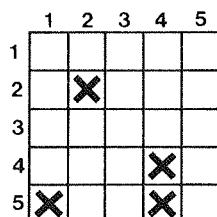
Sprungziel	Adresse	Eingabe-Befehl	Mnemonic	Sprung nach
	2D	<b>E3C</b>	BRZ 3C	R4
	2E	<b>886</b>	CMP 8,6	
	2F	<b>E31</b>	BRZ 31	
	30	<b>C33</b>	GOTO 33	
	31	<b>897</b>	CMP 9,7	
	32	<b>E3E</b>	BRZ 3E	R6
	33	<b>1FC</b>	MOVI F,C	
	34	<b>FEC</b>	DOT C	
	35	<b>10C</b>	MOVI 0,C	
	36	<b>FEC</b>	DOT C	
	37	<b>C18</b>	GOTO 18	
R0	38	<b>100</b>	MOVI 0,0	
	39	<b>C40</b>	GOTO 40	
R2	3A	<b>102</b>	MOVI 0,2	
	3B	<b>C40</b>	GOTO 40	
R4	3C	<b>104</b>	MOVI 0,4	
	3D	<b>C40</b>	GOTO 40	
R6	3E	<b>106</b>	MOVI 0,6	
	3F	<b>1FC</b>	MOVI F,C	
	40	<b>1FA</b>	MOVI F,A	
	41	<b>FEA</b>	DOT A	
	42	<b>10A</b>	MOVI 0,A	
	43	<b>FEA</b>	DOT A	
	44	<b>71C</b>	SUBI 1,C	
	45	<b>E47</b>	BRZ 47	
	46	<b>C40</b>	GOTO 40	
	47	<b>51F</b>	ADDI 1,F	
	48	<b>94F</b>	CMPI 4,F	
	49	<b>E4E</b>	BRZ 4E	
	4A	<b>1AA</b>	MOVI A,A	
	4B	<b>1AB</b>	MOVI A,B	
	4C	<b>F48</b>	DISP 4,8	
	4D	<b>C19</b>	GOTO 19	
	4E	<b>10F</b>	MOVI 0,F	
	4F	<b>F03</b>	HxDZ	
	50	<b>F3D</b>	DISP 3,D	
	51	<b>FF0</b>	KIN 0	
	52	<b>C00</b>	GOTO 00	
RND	53	<b>F05</b>	RND	
	54	<b>90D</b>	CMPI 0,D	
	55	<b>E59</b>	BRZ 59	
	56	<b>95D</b>	CMPI 5,D	
	57	<b>D59</b>	BRC 59	
	58	<b>C5B</b>	GOTO 5B	
	59	<b>59D</b>	ADDI 9,D	
	5A	<b>C54</b>	GOTO 54	
	5B	<b>90E</b>	CMPI 0,E	
	5C	<b>E60</b>	BRZ 60	
	5D	<b>95E</b>	CMPI 5,E	
	5E	<b>D60</b>	BRC 60	
	5F	<b>C62</b>	GOTO 62	
	60	<b>53E</b>	ADDI 3,E	
	61	<b>C5B</b>	GOTO 5B	

Sprungziel	Adresse	Eingabe-Befehl	Mnemonic	Sprung nach
	62	<b>F07</b>	RET	
V3	63	<b>8D4</b>	CMP D,4	
	64	<b>E66</b>	BRZ 66	
	65	<b>C68</b>	GOTO 68	
	66	<b>8E5</b>	CMP E,5	
	67	<b>E74</b>	BRZ 74	STC
V2	68	<b>8D2</b>	CMP D,2	
	69	<b>E6B</b>	BRZ 6B	
	6A	<b>C6D</b>	GOTO 6D	
	6B	<b>8E3</b>	CMP E,3	
	6C	<b>E74</b>	BRZ 74	STC
V1	6D	<b>8D0</b>	CMP D,0	
	6E	<b>E70</b>	BRZ 70	
	6F	<b>C72</b>	GOTO 72	
	70	<b>8E1</b>	CMP E,1	
	71	<b>E74</b>	BRZ 74	STC
	72	<b>F0A</b>	RSC	
	73	<b>F07</b>	RET	
STC	74	<b>F09</b>	STC	
	75	<b>F07</b>	RET	

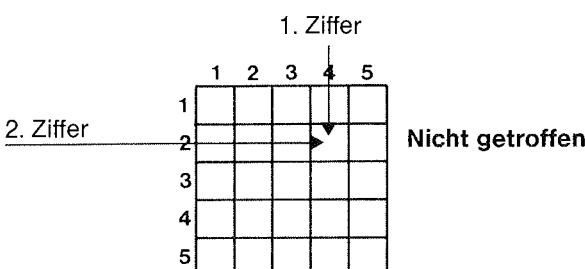
Piezo-Summer wieder anschließen, dann  
Programm-Start: HALT – NEXT – 00 – RUN.

Das Display bleibt 2 bis 3 Sekunden dunkel. Der Computer arbeitet, was an den blinkenden Carry- oder Zero-Flags zu sehen ist. Dann zeigt das Display: **00**.

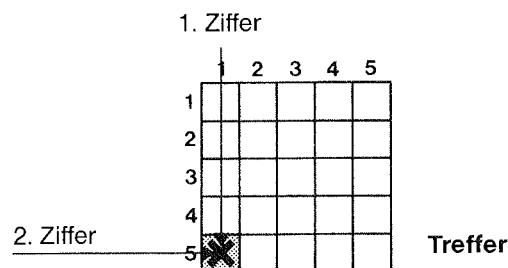
Der Computer hat inzwischen für uns nicht sichtbar, die 4 Schiffe auf dem Spielfeld verteilt. Die Verteilung erfolgt unter Einbeziehung des Zufallsgenerators. Die Schiffe sind also bei jedem Spiel in einer anderen Anordnung versteckt. Nehmen wir einmal an, daß sich die Schiffe an den im Spielplan (siehe Abb.) angekreuzten Positionen befinden.



Wir können zwei Ziffern eingeben. Die erste Ziffer gilt nur für die senkrechten Spielfeld-Nummern, die zweite Ziffer für die waagerechten Spielfeld-Nummern. Wir geben zum Beispiel **42** ein. Der Computer meldet sich mit einem kurzen Ton und zeigt die 42 an. Dieser Schuß war in unserem Beispiel kein Treffer.



Würden wir jetzt beispielsweise **15** eingeben, bringt der Computer einen Intervalton und das Display meldet: **AA15**. Das erste Schiff wurde versenkt. Würden wir anschließend nochmals die gleiche Zahlenkombination eingeben, wird an dieser Stelle kein Treffer mehr gemeldet, weil ja das Schiff bereits versenkt wurde.



Sobald auch das vierte Schiff getroffen wurde, meldet sich der Computer wieder mit dem Intervalton und auf dem Display werden 3 Stellen angezeigt. Wir können ablesen, wieviel Versuche notwendig waren, um alle 4 Schiffe zu treffen.

Nach Betätigung einer beliebigen Zahlen-Taste werden erneut 4 Schiffe versteckt, das Display zeigt wieder 00, ein neues Spiel kann beginnen.

#### Das interessiert den Programmierer:

Wir vergleichen die Beschreibung mit dem Programm-Ablaufplan und der Programm-Tabelle.

Das Spiel ist verhältnismäßig einfach zu programmieren, weil es keine spezielle Gewinn-Strategie gibt. Zuerst müssen die Schiffe verteilt werden. Zufällige Zahlen zwischen 1 und 5 werden in die Register 0 bis 7 gespeichert. Die Koordinaten des ersten Schiffes stehen in den Register 0 und 1, die des zweiten Schiffes in den Registern 2 und 3 usw.

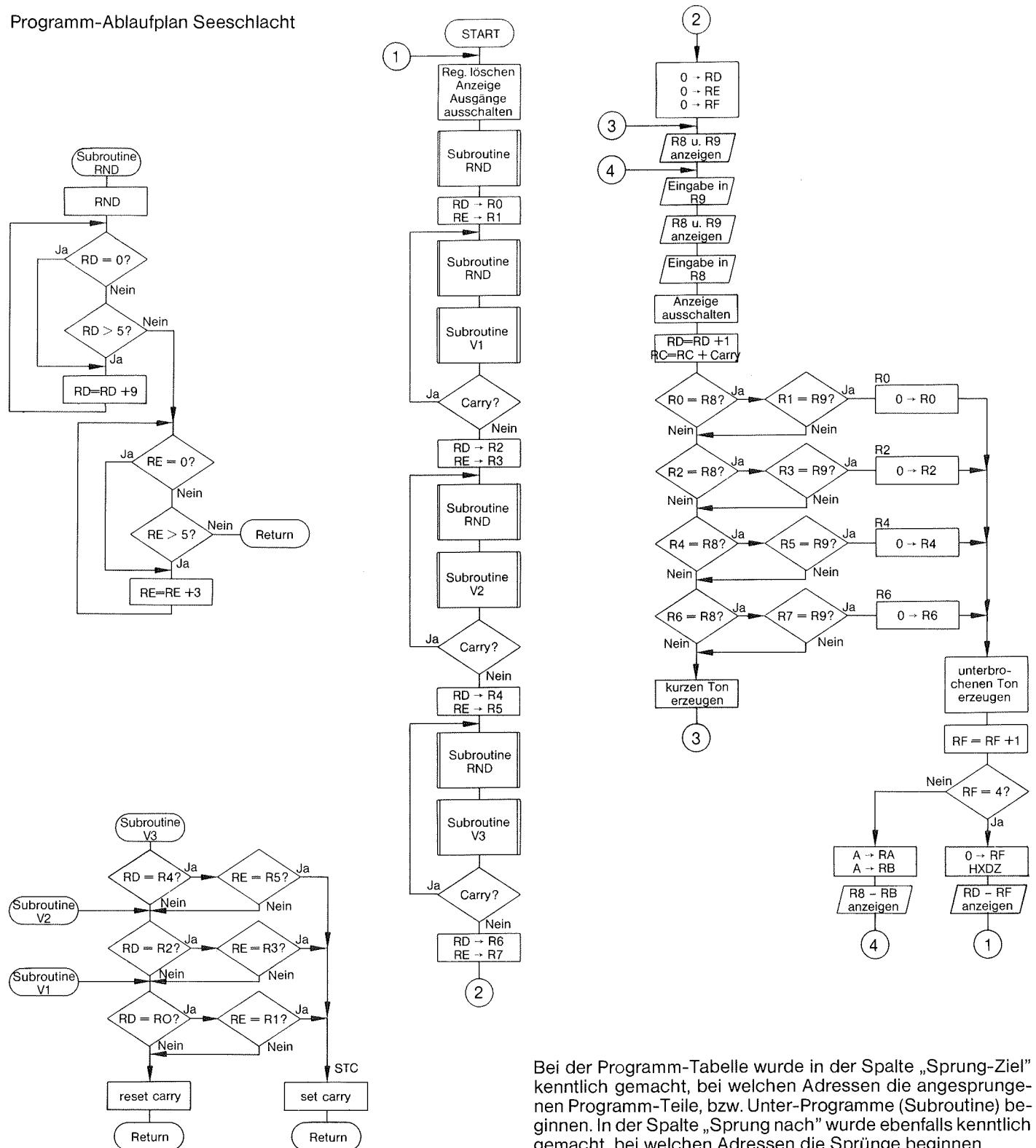
Durch die **Unter-Programme (Subroutine)** mit den Bezeichnungen **V1** (**V=VERGLEICHEN**), **V2** und **V3** wird kontrolliert, daß bei der Verteilung der Schiffe kein Feld doppelt belegt wird.

Anschließend werden die eingegebenen Koordinaten in den Registern 8 und 9 gespeichert. Durch Vergleichs-Befehle wird kontrolliert, ob die eingegebenen Koordinaten mit den Koordinaten eines versteckten Schiffes übereinstimmen.

Durch die **Subroutine RND** wird eine Zufallszahl in den Registern D, E und F gespeichert. Die Zufallszahlen in den Registern D und E ergeben die Koordinaten der versteckten Schiffe, weshalb nur die Zahlen zwischen 1 und 5 verwendbar sind. Durch Vergleiche wird kontrolliert, ob in diesen Registern eine Zufallszahl 0 oder größer als 5 erscheint. Wenn „ja“ wird solange der Wert 9 zum Register D (bzw. der Wert 3 zum Register E) addiert, bis beide Zufallszahlen Werte zwischen 1 und 5 haben.

#### Ablaufplan See-Schlacht (s. Seite 8)

## Programm-Ablaufplan Seeschlacht



Bei der Programm-Tabelle wurde in der Spalte „Sprung-Ziel“ kenntlich gemacht, bei welchen Adressen die angesprochenen Programm-Teile, bzw. Unter-Programme (Subroutine) beginnen. In der Spalte „Sprung nach“ wurde ebenfalls kenntlich gemacht, bei welchen Adressen die Sprünge beginnen.

# Code-Brecher

Code-Brecher sind Denk-Spiele. Die bekanntesten Arten derartiger Code-Spiele sind z. B. Master-Mind und Super-Hirn.

Bei unserem Code-Brecher muß ein 4-stelliger Zahlen-Code erraten werden.

Der Computer ermittelt mit dem integrierten Zufallsgenerator eine 4-stellige Zahlen-Kombination, welche unsichtbar gespeichert wird. Es müssen nicht nur die richtigen Zahlen, sondern die gesamte Zahlen-Kombination in der richtigen Reihenfolge erraten werden. Der Computer gibt alle Auskünfte, z. B. ob eine richtige Zahl gefunden wurde, ob sie an der richtigen Stelle steht usw. und er gibt am Spielende bekannt, wieviele Rateversuche notwendig waren.

Um dieses interessante Spiel ausprobieren zu können, geben wir nach HALT – NEXT – 00 das folgende Programm ein:

## Programm: Code-Brecher

Sprungziel	Adresse	Eingabe-Befehl	Mnemonic	Sprung nach
00	<b>F08</b>	CLEAR		
01	<b>F49</b>	DISP 4,9		
02	<b>FFC</b>	KIN C		
03	<b>FFB</b>	KIN B		
04	<b>FFA</b>	KIN A		
05	<b>FF9</b>	KIN 9		
06	<b>F02</b>	DISOUT		
07	<b>F05</b>	RND		
08	<b>4D9</b>	ADD D,9		
09	<b>4EA</b>	ADD E,A		
0A	<b>4FB</b>	ADD F,B		
0B	<b>4DC</b>	ADD D,C		
0C	<b>579</b>	ADDI 7,9		
0D	<b>999</b>	CMPI 9,9		
0E	<b>D0C</b>	BRC 0C		
0F	<b>57A</b>	ADDI 7,A		
10	<b>99A</b>	CMPI 9,A		
11	<b>D0F</b>	BRC 0F		
12	<b>8A9</b>	CMP A,9		
13	<b>E0F</b>	BRZ 0F		
14	<b>57B</b>	ADDI 7,B		
15	<b>99B</b>	CMPI 9,B		
16	<b>D14</b>	BRC 14		
17	<b>8B9</b>	CMP B,9		
18	<b>E14</b>	BRZ 14		
19	<b>8BA</b>	CMP B,A		
1A	<b>E14</b>	BRZ 14		
1B	<b>57C</b>	ADDI 7,C		
1C	<b>99C</b>	CMPI 9,C		
1D	<b>D1B</b>	BRC 1B		
1E	<b>8C9</b>	CMP C,9		
1F	<b>E1B</b>	BRZ 1B		
20	<b>8CA</b>	CMP C,A		
21	<b>E1B</b>	BRZ 1B		
22	<b>8CB</b>	CMP C,B		
23	<b>E1B</b>	BRZ 1B		
24	<b>10D</b>	MOVI 0,D		
25	<b>10E</b>	MOVI 0,E		
26	<b>10F</b>	MOVI 0,F		

Sprungziel	Adresse	Eingabe-Befehl	Mnemonic	Sprung nach
	27	<b>F61</b>	DISP 6,1	
	28	<b>FF0</b>	KIN 0	
	29	<b>9A0</b>	CMPI A,0	
	2A	<b>E33</b>	BRZ 33	„Vergl.“
	2B	<b>D31</b>	BRC 31	„Zahl“
	2C	<b>034</b>	MOV 3,4	
	2D	<b>023</b>	MOV 2,3	
	2E	<b>012</b>	MOV 1,2	
	2F	<b>001</b>	MOV 0,1	
	30	<b>C27</b>	GOTO 27	
„Zahl“	31	<b>F49</b>	DISP 4,9	
	32	<b>C60</b>	GOTO 60	
„Vergl.“	33	<b>F02</b>	DISOUT	
	34	<b>105</b>	MOVI 0,5	
	35	<b>106</b>	MOVI 0,6	
	36	<b>017</b>	MOV 1,7	
	37	<b>098</b>	MOV 9,8	
	38	<b>B65</b>	CALL 65	„SR“
	39	<b>0A8</b>	MOV A,8	
	3A	<b>B6A</b>	CALL 6,A	„ZR“
	3B	<b>0B8</b>	MOV B,8	
	3C	<b>B6A</b>	CALL 6,A	„ZR“
	3D	<b>0C8</b>	MOV C,8	
	3E	<b>B6A</b>	CALL 6,A	„ZR“
	3F	<b>027</b>	MOV 2,7	
	40	<b>B6A</b>	CALL 6,A	„ZR“
	41	<b>0B8</b>	MOV B,8	
	42	<b>B6A</b>	CALL 6,A	„ZR“
	43	<b>0A8</b>	MOV A,8	
	44	<b>B65</b>	CALL 65	„SR“
	45	<b>098</b>	MOV 9,8	
	46	<b>B6A</b>	CALL 6,A	„ZR“
	47	<b>037</b>	MOV 3,7	
	48	<b>B6A</b>	CALL 6,A	„ZR“
	49	<b>0A8</b>	MOV A,8	
	4A	<b>B6A</b>	CALL 6,A	„ZR“
	4B	<b>0B8</b>	MOV B,8	
	4C	<b>B65</b>	CALL 65	„SR“
	4D	<b>0C8</b>	MOV C,8	
	4E	<b>B6A</b>	CALL 6,A	„ZR“
	4F	<b>047</b>	MOV 4,7	
	50	<b>B65</b>	CALL 65	„SR“
	51	<b>0B8</b>	MOV B,8	
	52	<b>B6A</b>	CALL 6,A	„ZR“
	53	<b>0A8</b>	MOV A,8	
	54	<b>B6A</b>	CALL 6,A	„ZR“
	55	<b>098</b>	MOV 9,8	
	56	<b>B6A</b>	CALL 6,A	„ZR“
	57	<b>51D</b>	ADDI 1,D	
	58	<b>FBE</b>	ADC E	
	59	<b>FBF</b>	ADC F	
	5A	<b>946</b>	CMPI 4,6	
	5B	<b>E5D</b>	BRZ 5D	

Sprungziel	Adresse	Eingabe-Befehl	Mnemonic	Sprung nach
	5C	C27	GOTO 27	
	5D	1E0	MOVI E,0	
	5E	1E5	MOVI E,5	
	5F	F60	DISP 6,0	
	60	FF0	KIN 0	
	61	F03	HxDZ	
	62	F3D	DISP 3,D	
	63	FF0	KIN 0	
	64	C00	GOTO 00	
„SR“	65	878	CMP 7,8	
	66	E68	BRZ 68	
	67	C69	GOTO 69	
	68	516	ADDI 1,6	
	69	F07	RET	
„ZR“	6A	878	CMP 7,8	
	6B	E6D	BRZ 6D	
	6C	C6E	GOTO 6E	
	6D	515	ADDI 1,5	
	6E	F07	RET	

Programm-Start: HALT – NEXT – 00 – RUN.

Das Display zeigt: **0000**. Nun müssen wir dem Computer helfen, eine Zufalls-Zahlenkombination zu finden. Hierfür geben wir 4 beliebige Zahlen oder Buchstaben ein (diese Eingabe hat mit der später zu erratenden Zahlenkombination nichts zu tun). Das Display schaltet sich für 1 Sekunde ab – dann wird **000000** angezeigt. Der Computer hat die zu ratende Zahlenkombination ermittelt und unsichtbar gespeichert.

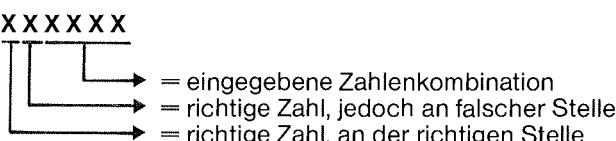
Für den Rate-Vorgang können wir immer 4 Zahlen zwischen 0 und 9 eingeben, beispielsweise **1 – 2 – 3 – 4**. Die eingegebenen Zahlen werden angezeigt. Durch Tastendruck **A** vergleicht der Computer die eingegebenen Zahlenwerte mit der unsichtbar gespeicherten Zahlenkombination. Das Display zeigt eine 6-stellige Anzeige, wobei auf den vier letzten Stellen die zuvor eingegebene Zahlenkombination (z. B. 1 – 2 – 3 – 4) weiterhin angezeigt wird. Werden auf den beiden linken Stellen z. B. **00** angezeigt, bedeutet dies, daß keine Zahl richtig erraten wurde.

Meldet das Display auf den beiden linken Anzeigestellen z. B. „**01**“, würde dies bedeuten:

**0** = keine richtig erratene Zahl an der richtigen Stelle.  
**1** = eine richtige Zahl, jedoch an der falschen Stelle.

Eine Meldung z. B. **21** auf den beiden linken Display-Stellen würde bedeuten, daß wir **2** richtige Ziffern an der richtigen Stelle erraten haben, und daß wir **1** weitere richtige Zahl erraten haben, die jedoch an der falschen Stelle steht. Nachdem wir die Zahlenkombination 1 – 2 – 3 – 4 eingegeben haben würde feststellen, daß hier von drei Zahlen richtig erraten wurden, wovon zwei Zahlen an der richtigen Stelle stehen, während eine Zahl an der falschen Stelle steht und eine Zahl in der vom Computer unsichtbar festgelegten Kombination noch erraten werden muß.

Die nachfolgende kleine Tabelle demonstriert uns nochmals wie das Display die Werte anzeigt:



Wir raten weiter. Vier neue Zahlen eingeben – Taste **A** betätigen. Sobald das Display auf den beiden linken Anzeigestellen richtige Zahlen meldet, sollten wir uns die eingegebenen Zahlenkombinationen aufschreiben, damit wir durch Verschieben und Neueingabe der einzelnen Stellen Schritt für Schritt die richtige Zahlenkombination erraten.

Der Code-Brecher ist ein interessantes Spiel und es erfordert viel logisches Denken, die richtige Zahlenkombination zu finden. Bei jeder Neueingabe nicht vergessen, die Taste A zu betätigen.

Würden wir nach einiger Zeit z. B. durch Eingabe der Zahlen 6 – 8 – 5 – 7 (dann Taste A) die richtige Zahlenkombination finden, meldet das Display: **E6857E**. Betätigten wir anschließend nochmals die Taste A zeigt das Display dreistellig, wieviel Versuche notwendig waren.

Wird erneut die Taste A betätigt, meldet das Display wie am Spielanfang: **0000** und sobald 4 beliebige Zahltasten betätigt wurden, kann ein neues Code-Brecher-Spiel beginnen.

Dauert uns der Spielverlauf zu lange, können wir uns mit der Taste **B** die vom Computer festgelegte Zahlenkombination anzeigen lassen. Das Spiel wird hierdurch allerdings abgebrochen und bei erneutem Tastendruck A wird auf dem Display die Anzahl der bisherigen Rateversuche angezeigt.

Das Programm muß sehr sorgfältig eingegeben werden, weil bei einer falschen Befehls-Eingabe die laufenden Berechnungen des Computers nicht stimmen und somit falsche Ergebnisse angezeigt werden. Wir können dies sehr leicht testen, wenn wir uns am Anfang nach einigen Rateversuchen durch die Taste B die zu erratende Zahlenkombination anzeigen lassen. Notfalls das Programm mit HALT – NEXT – 00 und weitere Betätigung der NEXT-Taste kontrollieren und richtigstellen.

**Das interessiert den Programmierer:**

Wir vergleichen wieder Programm-Tabelle und Ablaufplan. Der erste Programm-Teil (Adresse 00 bis 23) sorgt für die vier verschiedenen Zufallszahlen, welche später zu erraten sind. Obwohl wir mit Hilfe des im Mikroprozessors integrierten Zufallsgenerators diese Zufallszahlen ermitteln, ist es schwierig, effektiv vier unabhängige (voneinander verschiedene) Zufallszahlen zu erzielen. Angenommen, man würde lediglich durch 4 hintereinander gesetzte RND-Befehle die sich ergebenden Zufallszahlen in die 4 Register abspeichern, würden voneinander „abhängige“ Zahlen entstehen. Wäre beispielsweise die erste Zufallszahl die Ziffer 1, wären die nächsten 3 Ziffern von dieser Zahl abhängig, d. h. jedesmal, wenn sich als erste Zufallszahl die Ziffer 1 ergibt, würde sich die gleiche Zahlen-Kombination einstellen. Dies ergibt sich, weil die Ausführungszeiten der Befehle immer gleichlang sind und somit auch der „Zufallszähler“ in diesem gleichen Zeitabschnitt mit einer gleichen Schrittzahl weiterzählt.

Um vier voneinander unabhängige Zufallszahlen zu erhalten, wurde folgender Trick angewandt: Es werden 4 beliebige Ziffern eingegeben, wobei zu diesen jeweils eine Zufallszahl addiert wird. Anschließend wird kontrolliert, ob diese Zahlen kleiner als 9 sind. Ist dies nicht der Fall, wird solange der Wert 7 dazuaddiert, bis sich Werte ergeben, die 9 oder kleiner als 9 sind. Hierdurch werden die hexadezimalen Ziffern A bis F ausgeschieden.

Außerdem werden die vier Zufallszahlen überprüft, ob evtl. eine gleiche Zahl doppelt vorhanden ist. Falls ja, wird erneut der Wert 7 dazuaddiert, bis alle 4 Zufallszahlen unterschiedlich sind. Auf diese Weise werden vier verschiedene und nicht voneinander abhängige Zufallszahlen ermittelt.

Bei Adresse 28 erfolgt die Eingabe der „Rate-Zahlen“.

Bei Adresse 31 ist die Anzeige der Zufallszahlen möglich. Bei den Adressen 57 bis 59 wird die Anzahl der Rate-Versuche in den Registern D bis F hexadezimal mitgezählt. Der Vergleichsbefehl bei Adresse 5A vergleicht, ob Register 6 den Wert 4 enthält. Wenn ja, wurde die richtige Zahlenkombination erraten.

In der Programm-Tabelle finden wir in der ersten Spalte noch Hinweise, bei welchen Adressen die einzelnen Sprungziele beginnen, während in der letzten Spalte Hinweise vorhanden sind, bei welchen Adressen entsprechende Sprünge, z. B. zu den Unter-Programmen (Subroutine) ausgeführt werden.

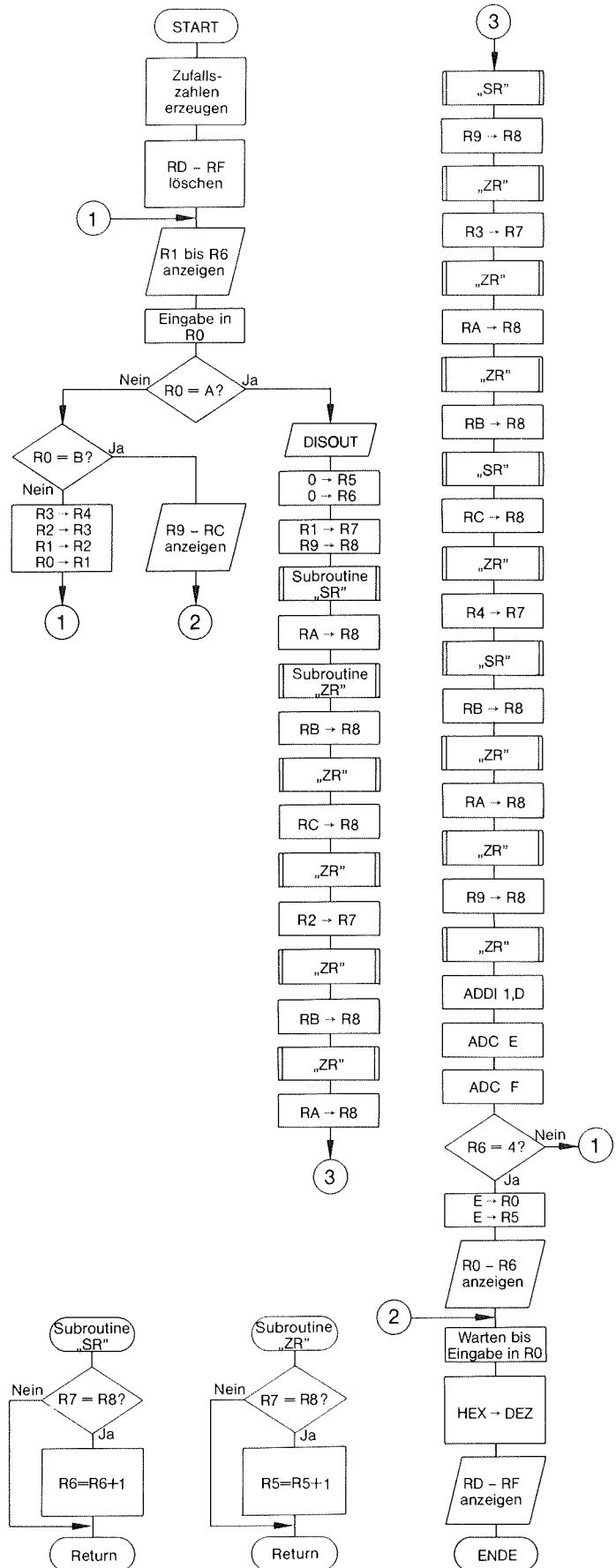
Aus dem Programm-Ablaufplan sind die einzelnen Arbeitsschritte ersichtlich. In den Registern 1 bis 4 wird die vom Spieler erdachte „Rate-Zahl“ eingegeben und anschließend wird jede einzelne der eingegebenen Ziffern mit den in den Registern 9 bis C gespeicherten Zufallszahlen des Computers verglichen. Hierdurch ergeben sich die Sprünge zur Subroutine „SR“ (= **S**telle **R**ichtig?), bzw. zur Subroutine **ZR** (= **Z**ahl **R**ichtig?). Die Subroutine „SR“ vergleicht, ob die richtige Zahl an der richtigen Stelle steht. Ist dies der Fall, wird im Register 6 der Wert 1 da-zuaddiert.

In der Subroutine „ZR“ wird verglichen, ob zwar eine richtige Zahl eingegeben wurde, die jedoch an der falschen Stelle steht. Wenn ja, wird in Register 5 der Wert 1 dazuaddiert.

Die Anzahl der Versuche wird in den Registern D, E und F hexadezimal mitgezählt. Erst wenn die Anzahl der Versuche angezeigt werden soll, wird das hexadezimale Ergebnis in dezimale Zahlen umgewandelt.

Es ist unmöglich, bei diesen längeren Programmen jeden einzelnen Programm-Schritt (wie im ersten Teil des Anleitungsbuches) zu erklären. Wer sich die Mühe macht, den Programm-Ablaufplan konsequent durchzuarbeiten, wird jedoch den Sinn und die Arbeitsweise des Programmes schnell verstehen.

## Programm-Ablaufplan: Code-Brecher



## „Nimm 2“

Im ersten Teil des Anleitungsbuches haben wir eine einfache Version des Nimm-Spiels (Festprogramm PGM 7) kennengelernt. Das folgende Programm ist eine vielfach interessantere Variante des Nimm-Spiels. Während wir bei der einfachen Ausführung nur ein Häufchen hatten, von welchem die Hölzchen wegzunehmen waren, so sind jetzt drei Häufchen vorhanden, wobei abwechselnd jeweils von einem Häufchen die Hölzchen weggenommen werden können. Es muß mindestens 1 Hölzchen, es darf aber auch ein ganzes Häufchen weggenommen werden. Gewinner ist derjenige, welcher das letzte (oder die letzten) Hölzchen wegnnehmen kann. Man beachte den Unterschied zu der früheren Nimm-Spiel-Version, wobei derjenige das Spiel verloren hat, welcher das letzte Hölzchen nehmen mußte.

Die neue Version „Nimm 2“ kann in drei verschiedenen Spielstärken ausgeführt werden. Um es vorweg zu sagen: In der Spielstärke 3 ist es fast unmöglich gegen den Computer zu gewinnen.

Bevor der Spielverlauf erklärt wird, zunächst nach HALT – NEXT – 00 das Programm lt. Tabelle eingeben.

### Programm: Nimm 2

Sprungziel	Adr. Nr.	Eingabe-Befehl	Mnemonic	Sprung nach	Erklärungen
00	<b>F08</b>	CLEAR			
01	<b>FE0</b>	DOT 0			
02	<b>F10</b>	DISP 1,0			
03	<b>FF8</b>	KIN 8			Eingabe der Spielstärke
04	<b>F61</b>	DISP 6,1			
05	<b>FF3</b>	KIN 3			Eingabe der Häufchengrößen
06	<b>FF2</b>	KIN 2			
07	<b>FF1</b>	KIN 1			
08	<b>FF6</b>	KIN 6			Von welchem Häufchen?
09	<b>FF5</b>	KIN 5			„wieviele?“
0A	<b>905</b>	CMPI 0,5			Null?
0B	<b>E23</b>	BRZ 23	„SCOMP“		Ja, dann Computer erster Spielzug
0C	<b>C12</b>	GOTO 12			
0D	<b>105</b>	MOVI 0,5			
0E	<b>FF6</b>	KIN 6			„welches Häufchen?“
0F	<b>F61</b>	DISP 6,1			
10	<b>FF5</b>	KIN 5			„wieviele?“
11	<b>E10</b>	BRZ 10			wenn Null, „wieviele?“ wiederholen
12	<b>916</b>	CMPI 1,6			Häufchen 1?
13	<b>E1E</b>	BRZ 1 E			
14	<b>926</b>	CMPI 2,6			Häufchen 2?
15	<b>E1A</b>	BRZ 1A			
16	<b>815</b>	CMP 1,5			Ist Eingabe größer als Häufchen 3?
17	<b>D0E</b>	BRC 0E			Ja, dann Neueingabe
18	<b>651</b>	SUB 5,1			
19	<b>C21</b>	GOTO 21	„SPGEW“		
1A	<b>825</b>	CMP 2,5			Eingabe größer als Häufchen 2?
1B	<b>D0E</b>	BRC 0E			Ja, dann Neueingabe
1C	<b>652</b>	SUB 5,2			
1D	<b>C21</b>	GOTO 21	„SPGEW“		
1E	<b>835</b>	CMP 3,5			Eingabe größer als Häufchen 1?

Sprungziel	Adr. Nr.	Eingabe-Befehl	Mnemonic	Sprung nach	Erklärungen
	1F	<b>D0E</b>	BRC 0E		Ja, dann Neueingabe
	20	<b>653</b>	SUB 5,3		
„SPGEW“	21	<b>B5C</b>	CALL 5C	„GEW“	Alle drei Häufchen Null?
	22	<b>D4E</b>	BRC 4E	„SPGEW“	Ja, dann hat der Spieler gewonnen
„SCOMP“	23	<b>F02</b>	DISOUT		Berechnung des Spielzuges des Computers
	24	<b>01D</b>	MOV 1,D		Häufchen 3 (Register 1) und Häufchen 2 werden in Reg. D und E abgespeichert u. ohne Übertrag addiert
	25	<b>02E</b>	MOV 2,E		
	26	<b>B66</b>	CALL 66	„XOR“	
	27	<b>8D3</b>	CMP D,3		Ergebnis (Reg. D) kleiner als Häufchen 1 (Reg. 3)?
	28	<b>E35</b>	BRZ 35	„-1“	(wenn gleich, dann Sprung nach -1)
	29	<b>D42</b>	BRC 42	„E1“	Ja, Sprung nach E1
	2A	<b>03D</b>	MOV 3,D		
	2B	<b>01E</b>	MOV 1,E		
	2C	<b>B66</b>	CALL 66	„XOR“	
	2D	<b>8D2</b>	CMP D,2		
	2E	<b>E35</b>	BRZ 35	„-1“	
	2F	<b>D48</b>	BRC 48	„E2“	
	30	<b>02D</b>	MOV 2,D		
	31	<b>03E</b>	MOV 3,E		
	32	<b>B66</b>	CALL 66	„XOR“	
	33	<b>8D1</b>	CMP D,1		
	34	<b>D4A</b>	BRC 4A	„E3“	
„-1“	35	<b>901</b>	CMPI 0,1		„Verlegenheitszug“, von einem der drei Häufchen eins wegnehmen
	36	<b>E39</b>	BRZ 39		
	37	<b>711</b>	SUBI 1,1		
	38	<b>C3E</b>	GOTO 3E	„CGEW“	
	39	<b>902</b>	CMPI 0,2		
	3A	<b>E3D</b>	BRZ 3D		
	3B	<b>712</b>	SUBI 1,2		
	3C	<b>C3E</b>	GOTO 3E	„CGEW“	
	3D	<b>713</b>	SUBI 1,3		
„CGEW“	3E	<b>B5C</b>	CALL 5C	„GEW“	Alle drei Häufchen Null?
	3F	<b>D59</b>	BRC 59	„CGEW“	Ja, dann hat Computer gewonnen
	40	<b>F31</b>	DISP 3,1		Andernfalls Häufchen neu anzeigen
	41	<b>C0D</b>	GOTO 0D		
„E1“	42	<b>918</b>	CMPI 1,8		bei Spielstärke 1 oder 2
	43	<b>E35</b>	BRZ 35	„-1“	(steht in Reg. 8) nur einen „Verlegenheitszug“ ausführen
	44	<b>928</b>	CMPI 2,8		
	45	<b>E35</b>	BRZ 35	„-1“	
	46	<b>0D3</b>	MOV D,3		Andernfalls ist RD das Ergebnis
	47	<b>C3E</b>	GOTO 3E		
„E2“	48	<b>0D2</b>	MOV D,2		RD ist das Ergebnis (Häufchen 2)
	49	<b>C3E</b>	GOTO 3E		
„E3“	4A	<b>918</b>	CMPI 1,8		Spielstärke 1?
	4B	<b>E35</b>	BRZ 35	„-1“	Ja, dann Verlegenheitszug („-1“)
	4C	<b>0D1</b>	MOV D,1		Andernfalls ist Reg. D Ergebnis (Häufchen 3)
	4D	<b>C3E</b>	GOTO 3E		

Sprungziel	Adr. Nr.	Eingabe-Befehl	Mnemonic	Sprung nach	Erklärungen
SPGEW	4E	<b>1F0</b>	MOVI F,0		Anzeige: Spieler hat gewonnen
	4F	<b>FE1</b>	DOT 1		
	50	<b>519</b>	ADDI 1,9		
	51	<b>E54</b>	BRZ 54		
	52	<b>FE0</b>	DOT 0		
	53	<b>C4F</b>	GOTO 4F		
	54	<b>1E5</b>	MOVI E5		
	55	<b>1E6</b>	MOVI E6		
	56	<b>F61</b>	DISP 6,1		
	57	<b>FF0</b>	KIN 0		
	58	<b>C00</b>	GOTO 00		
„CGEW“	59	<b>1F0</b>	MOVI F,0		Anzeige: Computer hat gewonnen
	5A	<b>FE0</b>	DOT 0		
	5B	<b>C54</b>	GOTO 54		
„GEW“	5C	<b>901</b>	CMPI 0,1		Unterprogramm „GEW“ vergleicht, ob alle drei Häufchen Null sind
	5D	<b>D64</b>	BRC 64		
	5E	<b>902</b>	CMPI 0,2		
	5F	<b>D64</b>	BRC 64		
	60	<b>903</b>	CMPI 0,3		
	61	<b>D64</b>	BRC 64		
	62	<b>F09</b>	STC		
	63	<b>F07</b>	RET		
	64	<b>F0A</b>	RSC		
	65	<b>F07</b>	RET		
„XOR“	66	<b>0EC</b>	MOV E,C		Unterprogramm „XOR“
	67	<b>2DC</b>	AND D,C		
	68	<b>AED</b>	OR E,D		
	69	<b>F8D</b>	INV D		
	6A	<b>ACD</b>	OR C,D		
	6B	<b>F8D</b>	INV D		
	6C	<b>F07</b>	RET		

Nun sollte noch der Piezo-Summer wie üblich angeschlossen werden. Dann Programm-Start: HALT – NEXT – 00 – RUN.

Das Display zeigt **0**. Der Computer will die Spielstärke wissen – wir geben z. B. ein **1**. Das Display zeigt jetzt: **000000**.

Es ist festzulegen, wieviel Hölzchen auf den drei Häufchen liegen sollen. Beispiel: Häufchen Nr. 1 hat 3 Hölzchen (**3** eingeben). Häufchen-Nr. 2 hat 4 Hölzchen (**4** eingeben) und Häufchen-Nr. 3 hat 5 Hölzchen (**5** eingeben). Das Display zeigt jetzt **000345**.

Wir möchten vom zweiten Häufchen 1 Hölzchen wegnehmen (Eingabe **2 1**). Das Display müßte jetzt anzeigen: **335** – tatsächlich zeigt jedoch das Display an: **334**, weil der Computer ebenfalls 1 Hölzchen und zwar vom dritten Häufchen weggenommen hat.

Wir nehmen jetzt vom dritten Häufchen die verbleibenden 4 Hölzchen weg (Eingabe **3 4**). Die Anzeige müßte lauten: **330**. Nachdem der Computer jedoch vom zweiten Häufchen ebenfalls 1 Hölzchen wegnimmt, zeigt das Display tatsächlich: **320**.

Wir nehmen jetzt vom ersten Häufchen 1 Hölzchen (Eingabe **1 1**) und unter Berücksichtigung, daß der Computer gleichzeitig vom zweiten Häufchen ebenfalls 1 Hölzchen wegnimmt, zeigt das Display: **210**. Da auch der Computer immer nur von einem Häufchen etwas wegnehmen kann, richten wir es so ein, daß auf dem verbleibenden ersten und zweiten Häufchen nach

unserer Wegnahme jeweils noch 1 Hölzchen übrigbleibt. Also nehmen wir von den 2 Hölzchen des ersten Häufchens 1 Hölzchen weg (Eingabe **1 1**). Der Computer nimmt vom zweiten Häufchen das dort verbleibende letzte Hölzchen weg und das Display zeigt: **100**. Nachdem wir nun das letzte Hölzchen wegnehmen können, sind wir Sieger, das Display meldet: **EE0000** und der Piezo-Summer gibt einen Intervallton, welcher nach kurzer Zeit automatisch abgeschaltet wird.

Sobald eine beliebige Zahlentaste betätigt wird, erscheint auf dem Display wieder 0 – ein neues Spiel kann beginnen.

Wir können eine neue Spielstärke wählen (Eingabe: 1 oder 2 oder 3). Die jetzt einzugebende Häufchengröße muß mindestens 1 oder höchstens 9 betragen. Bei jedem Zug dürfen beliebig viele Hölzchen, aber immer nur von einem Häufchen weggenommen werden. Beim nächsten Zug kann die Wegnahme bei einem anderen Häufchen vorgenommen werden. Wenn der Computer das Spiel beginnen soll (die ersten Hölzchen wegnehmen soll) geben wir nach Eingabe der Häufchengröße ein: 00. Die folgende 3-stellige Display-Anzeige gibt die drei verbleibenden Häufchengrößen für unsere Eingabe (wie beschrieben) an.

Gewinnt der Computer (weil wir ihm die Chance geboten haben das oder die letzten Hölzchen wegzunehmen) bringt das Display ebenfalls die Meldung: **EE0000**. Gleichzeitig ertönt solange ein Dauerton, bis durch Tastenbetätigung wieder der Spielanfang erreicht ist. Für einen neuen Spielbeginn immer zuerst die Spielstärke eingeben.

Wenn wir eine falsche Eingabe machen (z. B. von einem Häufchen mehr wegnehmend als Hölzchen vorhanden sind) merkt der Computer, daß gemogelt wurde – das Programm bleibt stehen. Wir geben nochmals (jedoch ohne zu mogeln) die Häufchen-Nummer und die Anzahl der wegzunehmenden Hölzchen ein. Der Programmablauf kann fortgesetzt werden. Oder mit HALT – NEXT – 00 das Programm neu starten.

Wenn wir bei Spielbeginn die Spielstärke 2 eingeben, müssen wir uns schon etwas mehr konzentrieren, das Spiel zu gewinnen. Bei Eingabe der Spielstärke 3 wird es uns kaum gelingen, den Computer zu besiegen. Hier wird sehr deutlich, daß wir durch eine intelligente Programmierung den Computer soweit bringen können, daß er uns mit unseren eigenen Waffen schlägt. Eine interessante Perspektive, z. B. in Erinnerung an Science-Fiction-Filme, bei welchen von Menschen programmierte Computer und Roboter die intelligente Menschheit besiegen.

### Das interessiert den Programmierer:

Bei der Programm-Tabelle wurden in der ersten Spalte wieder die Adressen kenntlich gemacht, bei welchen die Sprung-Ziele beginnen. In der vorletzten Spalte (neben Erklärungen) sind die Adressen erkenntlich, von welchen aus zu den Sprung-Zielen gesprungen wird. Hierfür werden einige Abkürzungen verwendet:

„-1“ = „Verlegenheitszug“ ausführen

„CGEW?“ = Computer gewinnt?

„SPGEW“ = Spieler gewinnt

„XOR“ = Unter-Programm „XOR“

„GEW“ = Unter-Programm Gewinnberechnung

„E1“, „E2“, „E3“ = Eingabe der Spielstärke (1 oder 2 oder 3).

„SCOMP“ = Spielzug des Computers

Für das Spiel „Nimm 2“ gibt es ähnlich wie bei der einfachen Nimm-Spiel-Version eine „optimale Gewinn-Strategie“, welche es dem Spieler oder dem Computer ermöglicht zu gewinnen. Ein Unentschieden gibt es nicht.

Wie beim Nimm-Spiel gibt es „Gewinn-Zahlen“ oder „sichere Positionen“. Typisches Beispiel hierfür sind die sogenannten „Pärchen“ wie z. B. bei der Display-Anzeige: **033** (33 = Pärchen). Der Spielstand wäre bei diesem Beispiel: Erstes Häufchen **0**, zweites Häufchen **3** und drittes Häufchen ebenfalls **3**. Bei dieser Situation hat der Gegenspieler folgende drei Möglichkeiten:

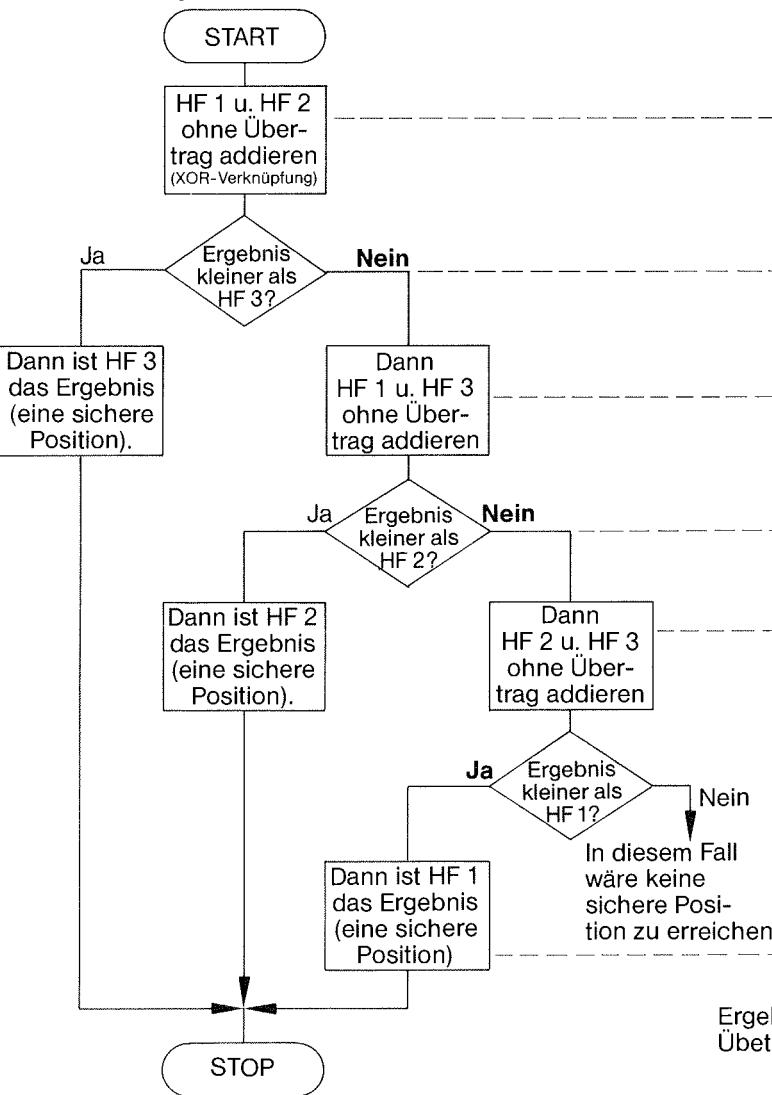
1. Von **033** verbleibt nach Wegnahme **032** oder **023**.
2. Von **033** verbleibt nach Wegnahme **031** oder **013**
3. Von **033** verbleibt nach Wegnahme **030** oder **003**

Aus der ersten und zweiten Möglichkeit kann wieder ein neues „Pärchen“ gebildet werden. Bei der dritten Möglichkeit wird der Spieler gewinnen, weil der Gegenspieler nur noch auf einem Häufchen 3 Hölzchen übrig lässt, die für den letzten Zug (also für den Gewinner) verbleiben.

Für den Computer gibt es eine relativ einfache Art festzustellen, ob er eine Gewinn-Position erreicht hat. Wir erinnern uns an das Kapitel „logische Operationen“, in welchem demonstriert wurde, daß mit den AND- und OR-Befehlen die einzelnen Bits innerhalb des dualen Zahlensystems verändert werden können.

#### Vereinfachter Programm-Ablaufplan zur Berechnung einer „sicheren Position“

**HF** = Häufchengröße



ten. Gehen wir davon aus, daß auf den drei Häufchen die Werte **1 – 2 – 3** noch vorhanden sind, sieht das dual dargestellt folgendermaßen aus:

$$\begin{array}{r}
 1 = \text{dual } 0001 \\
 + 2 = \text{dual } 0010 \\
 + 3 = \text{dual } 0011
 \end{array}$$

Eine AND-/OR-Befehlskombination erbringt **dual 0000**

Bei vorstehendem Beispiel wurden die Spalten der einzelnen Dualzahlen (ohne Übertrag) „addiert“, wodurch sich ein Gesamtergebnis **0000** ergibt. Das Gesamtergebnis **0000** bedeutet eine „sichere Position“!

Ein anderes Beispiel: Es verbleiben auf den drei Häufchen noch folgende Hölzchen: **033**. Dual dargestellt ergibt sich folgende Situation:

$$\begin{array}{r}
 0 = \text{dual } 0000 \\
 + 3 = \text{dual } 0011 \\
 + 3 = \text{dual } 0011
 \end{array}$$

Eine AND-/OR-Befehlskombination erbringt **dual 0000**

Beim Start sind die HF: **2 – 4 – 5**

somit ist  
HF 1 = 2 Hölzchen  
HF 2 = 4 Hölzchen  
HF 3 = 5 Hölzchen

Rechenbeispiele:  
Dezimal      Dual

$$\begin{array}{r}
 \text{HF 1} = 2 \quad 0010 \\
 \text{HF 2} = +4 \quad 0100 \\
 \hline
 \text{Ergebnis} = 6 \quad 0110
 \end{array}$$

→ **Nein!** Nachdem das Ergebnis 6 größer als HF 3 (5 Hölzchen) ist, weiter rechnen:

$$\begin{array}{r}
 \text{HF 1} = 2 \quad 0010 \\
 \text{HF 3} = +5 \quad 0101 \\
 \hline
 \text{Ergebnis} = 7 \quad 0111
 \end{array}$$

→ **Nein!** Nachdem das Ergebnis 7 größer als HF 2 (4 Hölzchen) ist, weiter rechnen:

$$\begin{array}{r}
 \text{HF 2} = 4 \quad 0100 \\
 \text{HF 3} = +5 \quad 0101 \\
 \hline
 \text{Ergebnis} = 1 \quad 0001
 \end{array}$$

**Ja!** Das Ergebnis 1 ist größer als HF 1 (2 Hölzchen). Es ergibt sich eine sichere Position, wenn Computer von HF 1 ein Hölzchen wegnimmt, weil dann:

$$\begin{array}{r}
 \text{HF 1} = 1 \quad 0001 \\
 \text{HF 2} = 4 \quad 0100 \\
 \text{HF 3} = 5 \quad 0101
 \end{array}$$

Ergebnis ohne Übertrag addiert = **(1)0 0000**

Bei einer „unsicheren Position“ wäre das Ergebnis nicht 0000. Dies soll mit den Häufchengrößen: **2 – 4 – 5** ebenfalls dual dargestellt werden:

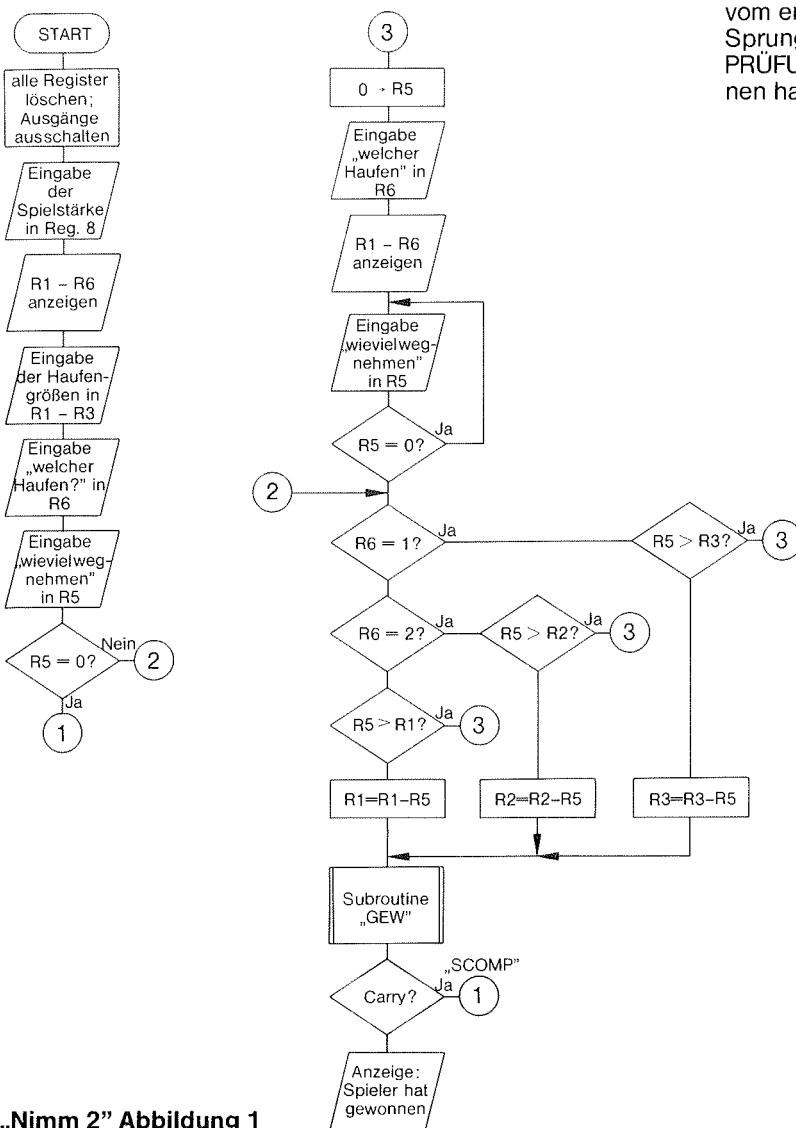
$$\begin{array}{r} 2 = \text{dual } 0010 \\ + 4 = \text{dual } 0100 \\ + 5 = \text{dual } 0101 \\ \hline \end{array}$$

Eine AND-/OR-Befehlskombination erbringt      dual 0011

Aus den Beispielen sehen wir, daß sich bei den dualen Werten bei einer „Addition“ folgendes ergibt:  $0 + 0 + 0 = 0$ . Oder:  $0 + 1 + 1 = 0$ . Oder (siehe drittes Beispiel mit den Häufchengrößen  $2 - 4 - 5$ ):  $1 + 0 + 0 = 1$ . Oder:  $0 + 0 + 1 = 1$ . Dreimal 0 addiert ergibt 0. 0 und zweimal 1 addiert ergibt ebenfalls 0. Jedoch zweimal 0 + 1 addiert ergibt 1.

Die dualen Zahlenstellen wurden „ohne Übertrag addiert“. Eine Addition ohne Übertrag ist eine sogenannte **XOR**- (exclusive **or**)-Verknüpfung, welche durch mehrere AND- und OR-Befehle erreicht werden kann.

Wir haben in einem vereinfachten Programm-Ablauf dargestellt, welche Rechenoperationen der Computer ausführen muß, um zu einer „sicheren Position“ zu kommen. Rechts neben dem Ablaufplan sind die Rechenbeispiele dezimal und dual dargestellt. Wir gehen davon aus, daß die drei Häufchengrößen die Werte (Anzahl der Hölzchen) haben: **2 4 5**.



Bei Eingabe der Spielstärke 1 war es leicht möglich gegen den Computer zu gewinnen. Aus der vorangegangenen Darstellung wissen wir, daß der Computer immer versucht, eine „sichere Position“ zu errechnen, indem er „Pärchen“ bildet. Geben wir bei Spielstärke 3 z.B. Häufchengrößen ein, bei welchen sich ein Pärchen ergibt (z. B. **0 6 6**) ist es nicht mehr möglich, gegen den Computer zu gewinnen. Probieren wir es einmal!

Erkenntnis: Eine „sichere Position“ für den Spieler gegen den Computer kann nicht erreicht werden, wenn bereits beim Spielbeginn (oder beim Verlauf des Spiels) der Computer eine „sichere Position“ durch „Pärchen-Bildung“ erreicht hat. Solange der Computer keine „sichere Position“ erreicht, macht er einen „Verlegenheitszug“, indem er von einem der drei Häufchen nur 1 Hölzchen wegnimmt in der Hoffnung, daß der Gegenspieler nicht aufpaßt, um während des Spielverlaufs dann doch noch eine „sichere Position“ zu erreichen.

Um den gesamten Programm-Ablauf besser zu verstehen, betrachten wir uns den Programm-Ablaufplan in einzelnen Abschnitten.

Die Abbildung 1 zeigt auf der linken Seite die Eingabe der Spielstärke und die Eingabe der Häufchengrößen. Auf der rechten Seite wird demonstriert was geschieht, sobald ein Spieler von einem Häufchen ein Hölzchen wegnimmt. Es wird vor allem geprüft, ob die weggenommene Hölzchenzahl überhaupt zulässig ist (z. B. ob 0 weggenommen wird, oder ob von einem Häufchen mehr Hölzchen weggenommen werden, als vorhanden sind). Anschließend wird die entsprechende Hölzchenzahl vom entsprechenden Häufchen abgezogen und es erfolgt ein Sprung zur **Subroutine** (Unter-Programm) „**GEW**“ (= GEWINN-PRÜFUNG), wobei dort überprüft wird, ob der Spieler gewonnen hat.

„Nimm 2“ Abbildung 1

Die Abbildung 2 zeigt die Berechnung der Spielzüge des Computers. Der Ablaufplan ist ähnlich aufgebaut wie bei der „Berechnung einer sicheren Position“. Zusätzlich sind in diesem Programm-Teil die Abfragen eingebaut, welche Spielstärke gewählt wurde (Register 8 = 1? bzw. Register 8 = 2?). Wurde z. B. die Spielstärke 1 oder 2 eingegeben, führt der Computer nicht

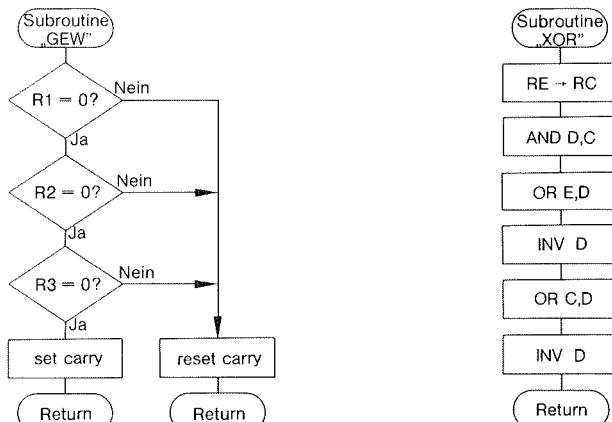
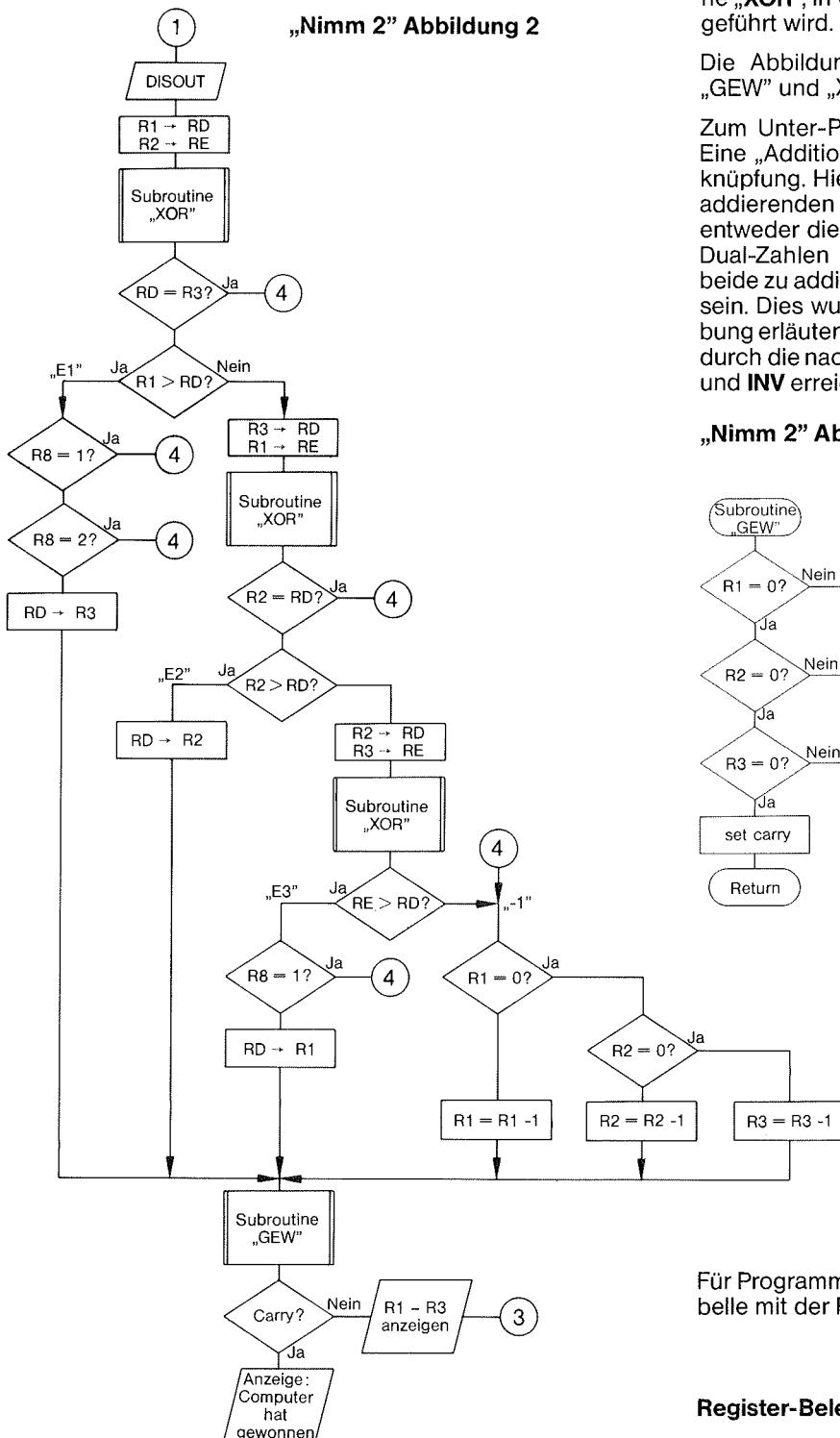
immer den „optimalen Zug“, sondern einen „Verlegenheitszug“ aus. Der Verlegenheitszug wird im Programm-Teil „-1“ bei der Programm-Übergangsstelle ④ berechnet, indem von einem Register, welches nicht den Wert 0 hat, der Wert 1 abgezogen wird.

In Abbildung 2 erkennen wir auch den Hinweis auf die Subroutine „XOR“, in welcher die duale Addition ohne Übertrag durchgeführt wird.

Die Abbildung 3 zeigt die Unter-Programme: Subroutine „GEW“ und „XOR“.

Zum Unter-Programm „XOR“ soll noch festgestellt werden: Eine „Addition ohne Übertrag“ ist eine sogenannte XOR-Verknüpfung. Hierdurch wird erreicht, daß das Ergebnis einer zu addierenden Dual-Stelle nur dann den Wert 1 ergibt, wenn entweder die eine oder die andere Stelle der zu addierenden Dual-Zahlen 1 ist. Sind beide zu addierenden Stellen 0 dann muß auch das Ergebnis 0 sein. Dies wurde bereits am Anfang der Programm-Beschreibung erläutert. Die XOR-Verknüpfung wird im Unter-Programm durch die nacheinander folgenden logischen Befehle **AND**, **OR** und **INV** erreicht.

„Nimm 2“ Abbildung 3



Für Programm-Spezialisten dürfte auch die nachstehende Tabelle mit der Register-Belegung nicht uninteressant sein:

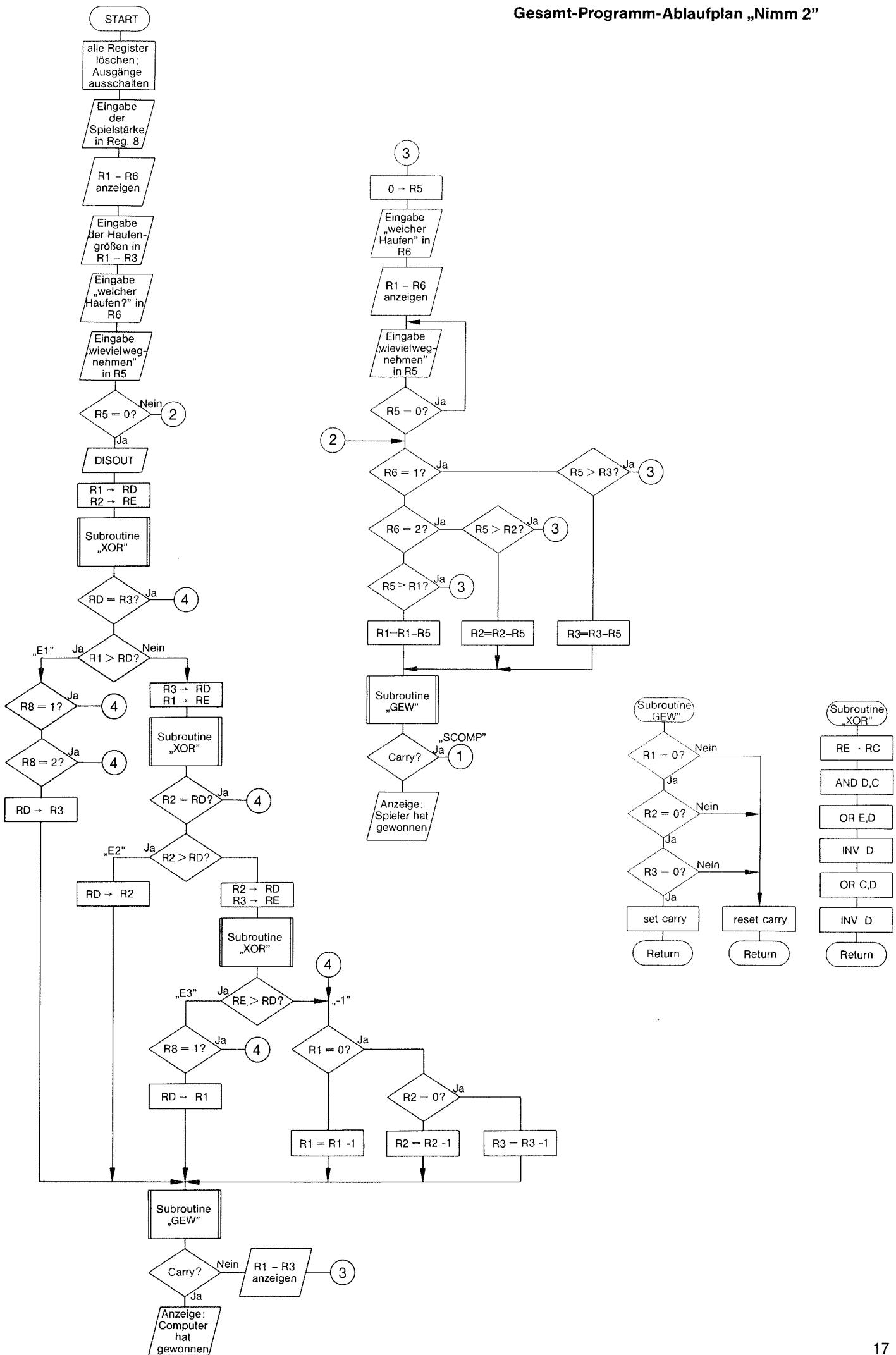
#### Register-Belegung „Nimm 2“

HF = Häufchengröße

#### Registerbelegung „Nimm 2“

Register-Nr.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Register-Inhalt	frei	HF 3	HF 2	HF 1	frei	Wieviel? von welchem HF?	frei	frei	frei	frei	frei	frei	„XOR“	„XOR“	„XOR“	frei

# Gesamt-Programm-Ablaufplan „Nimm 2“



# Morsezeichen-Decoder

Nach HALT – NEXT – 00 wird das große Programm sehr sorgfältig eingegeben. Der vom vorangegangene Versuch noch angeschlossene Piezo-Summer wird während des Programmierens abgeklemmt und nach der Programm-Eingabe wieder angeschlossen. Außerdem ist eine Verbindungsleitung zwischen der Buchse TAKT/CLOCK und EINGANG 4 herzustellen.

Bei der Programmierung fällt uns auf, daß z. B. bei Adresse 1D und 1E der NOP-Befehl verwendet wird, und daß vor allem in der zweiten Programm-Hälfte mehrfach hintereinander die gleichen Befehle einzugeben sind. Diese mehrfachen, gleichen Befehle sind erforderlich. Eine Erklärung finden wir im sich anschließenden Kapitel „Morse-Trainer“.

## Programm: Morsezeichen-Dekoder (Morse-Trainer)

Sprungziel	Adresse	Eingabe-Befehl	Mnemonic	Sprung nach	Erklärungen
	00	<b>F01</b>	NOP		
	01	<b>F02</b>	DISOUT		
	02	<b>F08</b>	CLEAR		
	03	<b>FE0</b>	DOT 0		
	04	<b>FF1</b>	KIN1		
	05	<b>9D1</b>	CMPI D,1		
	06	<b>D29</b>	BRC 29	„BUCHS“	„BUCHS“ = Buchstaben
	07	<b>9A1</b>	CMPI A,1		
	08	<b>EA2</b>	BRZ A2	„PUNKT“	
	09	<b>9B1</b>	CMPI B,1		
	0A	<b>EA9</b>	BRZ A9	„KOMMA“	
	0B	<b>9C1</b>	CMPI C,1		
	0C	<b>EBE</b>	BRZ BE	„AUTO“	
	0D	<b>DB0</b>	BRC B0	„SOS“	
	0E	<b>155</b>	MOVI 5,5		
	0F	<b>951</b>	CMPI 5,1		
	10	<b>D1F</b>	BRC 1F	„>5“	Eingegebene Zahl größer oder kleiner als 5?
	11	<b>901</b>	CMPI 0,1		
	12	<b>E18</b>	BRZ 18		
	13	<b>615</b>	SUB 1,5		
	14	<b>BD7</b>	CALL D7	„*“	
	15	<b>711</b>	SUBI 1,1		
	16	<b>E18</b>	BRZ 18		
	17	<b>C14</b>	GOTO 14		
	18	<b>905</b>	CMPI 0,5		
	19	<b>E00</b>	BRZ 00		
	1A	<b>BE3</b>	CALL E3	“–“	Erzeugung des Morsezeichens der Ziffern 1 bis 5
	1B	<b>715</b>	SUBI 1,5		
	1C	<b>C18</b>	GOTO 18		
	1D	<b>F01</b>	NOP		
	1E	<b>F01</b>	NOP		
„>5“	1F	<b>751</b>	SUBI 5,1		
	20	<b>615</b>	SUB 1,5		
	21	<b>BE3</b>	CALL E3	“–“	
	22	<b>711</b>	SUBI 1,1		
	23	<b>E25</b>	BRZ 25		
	24	<b>C21</b>	GOTO 21		
	25	<b>BD7</b>	CALL D7	“–“	
	26	<b>715</b>	SUBI 1,5		
	27	<b>E00</b>	BRZ 00		
	28	<b>C25</b>	GOTO 25		

„BUCHS“	29	<b>FF2</b>	KIN 2		
	2A	<b>9F1</b>	CMPI F,1		
	2B	<b>E46</b>	BRZ 46	„BU 2“	Bildung des HilfsCodes für die Buchstaben
	2C	<b>902</b>	CMPI 0,2		
	2D	<b>E60</b>	BRZ 60	„A“	
	2E	<b>912</b>	CMPI 1,2		
	2F	<b>E64</b>	BRZ 64	„C“	
	30	<b>922</b>	CMPI 2,2		
	31	<b>E68</b>	BRZ 68	„E“	
	32	<b>932</b>	CMPI 3,2		
	33	<b>E6C</b>	BRZ 6C	„G“	
	34	<b>942</b>	CMPI 4,2		
	35	<b>E70</b>	BRZ 70	„I“	
	36	<b>952</b>	CMPI 5,2		
	37	<b>E74</b>	BRZ 74	„K“	
	38	<b>962</b>	CMPI 6,2		
	39	<b>E78</b>	BRZ 78	„M“	
	3A	<b>972</b>	CMPI 7,2		
	3B	<b>E7C</b>	BRZ 7C	„O“	
	3C	<b>982</b>	CMPI 8,2		
	3D	<b>E80</b>	BRZ 80	„Q“	
	3F	<b>992</b>	CMPI 9,2		
	3F	<b>E84</b>	BRZ 84	„S“	
	40	<b>9A2</b>	CMPI A,2		
	41	<b>E88</b>	BRZ 88	„U“	
	42	<b>9B2</b>	CMPI B,2		
	43	<b>E8C</b>	BRZ 8C	„W“	
„Y“	44	<b>1B2</b>	MOVI B,2		
	45	<b>C96</b>	GOTO 96	„4“	
„BU 2“	46	<b>902</b>	CMPI 0,2		
	47	<b>E62</b>	BRZ 62	„B“	
	48	<b>912</b>	CMPI 1,2		
	49	<b>E66</b>	BRZ 66	„D“	
	4A	<b>922</b>	CMPI 2,2		
	4B	<b>E6A</b>	BRZ 6A	„F“	
	4C	<b>932</b>	CMPI 3,2		
	4D	<b>E6E</b>	BRZ 6E	„H“	
	4E	<b>942</b>	CMPI 4,2		
	4F	<b>E72</b>	BRZ 72	„J“	
	50	<b>952</b>	CMPI 5,2		
	51	<b>E76</b>	BRZ 76	„L“	
	52	<b>962</b>	CMPI 6,2		
	53	<b>E7A</b>	BRZ 7A	„N“	
	54	<b>972</b>	CMPI 7,2		
	55	<b>E7E</b>	BRZ 7E	„P“	
	56	<b>982</b>	CMPI 8,2		
	57	<b>E82</b>	BRZ 82	„R“	
	58	<b>992</b>	CMPI 9,2		
	59	<b>E86</b>	BRZ 86	„T“	
	5A	<b>9A2</b>	CMPI A,2		
	5B	<b>E8A</b>	BRZ 8A	„V“	
	5C	<b>9B2</b>	CMPI B,2		
	5D	<b>E8E</b>	BRZ 8E	„X“	
„Z“	5E	<b>9C2</b>	MOVI C,2		
	5F	<b>C96</b>	GOTO 96	„4“	

„A“	60	<b>142</b>	MOVI 4,2	
	61	<b>C92</b>	GOTO 92	„2“
„B“	62	<b>182</b>	MOVI 8,2	
	63	<b>C96</b>	GOTO 96	„4“
„C“	64	<b>1A2</b>	MOVI A,2	
	65	<b>C96</b>	GOTO 96	„4“
„D“	66	<b>182</b>	MOVI 8,2	
	67	<b>C94</b>	GOTO 94	„3“
„E“	68	<b>102</b>	MOVI 0,2	
	69	<b>C90</b>	GOTO 90	„1“
„F“	6A	<b>122</b>	MOVI 2,2	
	6B	<b>C96</b>	GOTO 96	„4“
„G“	6C	<b>1C2</b>	MOVI C,2	
	6D	<b>C94</b>	GOTO 94	„3“
„H“	6E	<b>102</b>	MOVI 0,2	
	6F	<b>C96</b>	GOTO 96	„4“
„I“	70	<b>102</b>	MOVI 0,2	
	71	<b>C92</b>	GOTO 92	„2“
„J“	72	<b>172</b>	MOVI 7,2	
	73	<b>C96</b>	GOTO 96	„4“
„K“	74	<b>1A2</b>	MOVI A,2	
	75	<b>C94</b>	GOTO 94	„3“
„L“	76	<b>142</b>	MOVI 4,2	
	77	<b>C96</b>	GOTO 96	„4“
„M“	78	<b>1C2</b>	MOVI C,2	
	79	<b>C92</b>	GOTO 92	„2“
„N“	7A	<b>182</b>	MOVI 8,2	
	7B	<b>C92</b>	GOTO 92	„2“
„O“	7C	<b>1E2</b>	MOVI E,2	
	7D	<b>C94</b>	GOTO 94	„3“
„P“	7E	<b>162</b>	MOVI 6,2	
	7F	<b>C96</b>	GOTO 96	„4“
„Q“	80	<b>1D2</b>	MOVI D,2	
	81	<b>C96</b>	GOTO 96	„4“
„R“	82	<b>142</b>	MOVI 4,2	
	83	<b>C94</b>	GOTO 94	„3“
„S“	84	<b>102</b>	MOVI 0,2	
	85	<b>C94</b>	GOTO 94	„3“
„T“	86	<b>182</b>	MOVI 8,2	
	87	<b>C90</b>	GOTO 90	„1“
„U“	88	<b>122</b>	MOVI 2,2	
	89	<b>C94</b>	GOTO 94	„3“
„V“	8A	<b>112</b>	MOVI 1,2	
	8B	<b>C96</b>	GOTO 96	„4“
„W“	8C	<b>162</b>	MOVI 6,2	
	8D	<b>C94</b>	GOTO 94	„3“
„X“	8E	<b>192</b>	MOVI 9,2	
	8F	<b>C96</b>	GOTO 96	„4“

„2“ =  
2 Tonfolgen

„1“	90	<b>111</b>	MOVI 1,1	
	91	<b>C97</b>	GOTO 97	
„2“	92	<b>121</b>	MOVI 2,1	
	93	<b>C97</b>	GOTO 97	
„3“	94	<b>131</b>	MOVI 3,1	
	95	<b>C97</b>	GOTO 97	
„4“	96	<b>141</b>	MOVI 4,1	
	97	<b>FA2</b>	SHL 2	
	98	<b>D9D</b>	BRC 9D	
	99	<b>BD7</b>	CALL D7	„•“
	9A	<b>711</b>	SUBI 1,1	
	9B	<b>E9F</b>	BRZ 9F	
	9C	<b>C97</b>	GOTO 97	
	9D	<b>BE3</b>	CALL E3	„—“
	9E	<b>C9A</b>	GOTO 9A	
	9F	<b>906</b>	CMPI 0,6	
	A0	<b>E00</b>	BRZ 00	
	A1	<b>CC7</b>	GOTO C7	„AUMO“
„PUNKT“	A2	<b>BD7</b>	CALL D7	„•“
	A3	<b>BE3</b>	CALL E3	„—“
	A4	<b>BD7</b>	CALL D7	„•“
	A5	<b>BE3</b>	CALL E3	„—“
	A6	<b>BD7</b>	CALL D7	„•“
	A7	<b>BE3</b>	CALL E3	„—“
	A8	<b>C9F</b>	GOTO 9F	
„KOMMA“	A9	<b>BE3</b>	CALL E3	„—“
	AA	<b>BE3</b>	CALL E3	„—“
	AB	<b>BD7</b>	CALL D7	„•“
	AC	<b>BD7</b>	CALL D7	„•“
	AD	<b>BE3</b>	CALL E3	„—“
	AE	<b>BE3</b>	CALL E3	„—“
	AF	<b>C9F</b>	GOTO 9F	
„SOS“	B0	<b>BD7</b>	CALL D7	„•“
	B1	<b>BD7</b>	CALL D7	„•“
	B2	<b>BD7</b>	CALL D7	„•“
	B3	<b>5FF</b>	ADDI F,F	
	B4	<b>5FF</b>	ADDI F,F	
	B5	<b>BE3</b>	CALL E3	„—“
	B6	<b>BE3</b>	CALL E3	„—“
	B7	<b>BE3</b>	CALL E3	„—“
	B8	<b>5FF</b>	ADDI F,F	
	B9	<b>5FF</b>	ADDI F,F	
	BA	<b>BD7</b>	CALL D7	„•“
	BB	<b>BD7</b>	CALL D7	„•“
	BC	<b>BD7</b>	CALL D7	„•“
	BD	<b>C00</b>	GOTO 00	
„AUTO“	BE	<b>116</b>	MOVI 1,6	
	BF	<b>F17</b>	DISP 1,7	

Erzeugung  
der Morse-  
zeichen für  
die Buchsta-  
ben aus dem  
Hilfscode

„AUMO“ =  
Automatisch  
Morse-  
zeichen  
bilden

Sprungziel	Adresse	Eingabe-Befehl	Mnemonic	Sprung nach	Erklärungen
	C0	<b>FF7</b>	KIN 7		
	C1	<b>F27</b>	DISP 2,7		
	C2	<b>FF8</b>	KIN 8		
	C3	<b>EC6</b>	BRZ C6		
	C4	<b>F60</b>	DISP 6,0		
	C5	<b>CC7</b>	GOTO C7		
	C6	<b>F02</b>	DISOUT		
„AUMO“	C7	<b>1FF</b>	MOVI F,F		Erzeugung der zufälligen Morsezeichen
	C8	<b>71F</b>	SUBI 1,F		
	C9	<b>ECB</b>	BRZ CB		
	CA	<b>CC8</b>	GOTO C8		
	CB	<b>4AD</b>	ADD A,D		
	CC	<b>87D</b>	BRC D5		
	CD	<b>DD5</b>	BRC D5		
	CE	<b>0D2</b>	MOV D,2		
	CF	<b>90E</b>	CMPI 0,E		
	D0	<b>ED3</b>	BRZ D3		
	D1	<b>1E1</b>	MOVI E,1		
	D2	<b>C2A</b>	GOTO 2A		
	D3	<b>1F1</b>	MOVI F,1		
	D4	<b>C2A</b>	GOTO 2A		
	D5	<b>57D</b>	ADDI 7,D		
	D6	<b>CCC</b>	GOTO CC		
„•“	D7	<b>1F4</b>	MOVI F,4		Unterprogramm für „•“
	D8	<b>FE4</b>	DOT 4		
	D9	<b>5FF</b>	ADDI F,F		
	DA	<b>5FF</b>	ADDI F,F		
	DB	<b>104</b>	MOVI 0,4		
	DC	<b>FE4</b>	DOT 4		
	DD	<b>FDE</b>	DIN E		
	DE	<b>EE1</b>	BRZ E1		
	DF	<b>53A</b>	ADDI 3,A		
	E0	<b>F07</b>	RET		
	E1	<b>5CA</b>	ADDI C,A		
	E2	<b>F07</b>	RET		
„—“	E3	<b>1F4</b>	MOVI F,4		Unterprogramm für „—“
	E4	<b>FE4</b>	DOT 4		
	E5	<b>714</b>	SUBI 1,4		
	E6	<b>EE8</b>	BRZ E8		
	E7	<b>CE5</b>	GOTO E5		
	E8	<b>FE4</b>	DOT 4		
	E9	<b>FDE</b>	DIN E		
	EA	<b>EEC</b>	BRZ EC		
	EB	<b>F8A</b>	INV A		
	EC	<b>F07</b>	RET		

Bei Betätigung der Taste 0 ertönt fünfmal ein langer Ton. Wird z. B. die Taste 7 betätigt, ergibt sich zweimal lang und dreimal kurz.

Wenn wir das Morsealphabet nicht gelernt haben, vergleichen wir mit der Alphabet-Tabelle und bei jeder Betätigung einer Zahlentaste bringt der Computer das entsprechende Morsezeichen.

### Morsezeichen

a	• —	n	— •	1	• — — — —
b	— • • •	o	— — —	2	• • — — —
c	— • — •	p	• — — •	3	• • • — —
d	— • •	q	— — • —	3	• • • • —
e	•	r	• — •	5	• • • • •
f	• • — •	s	• • • •	6	— • • • •
g	— — •	t	—	7	— — • • •
h	• • • •	u	• • —	8	— — — • •
i	• •	v	• • • —	9	— — — — •
j	• — — —	w	• — —	0	— — — — —
k	— • —	x	— • • —	Punkt	• — • — • —
l	• — • •	y	— • — —	Komma	— — • • — —
m	— —	z	— — • •	Notruf: SOS	• • • — — • • •

Selbstverständlich können wir nicht nur Zahlen, sondern auch sämtliche Buchstaben des Alphabets in Morsezeichen darstellen. Hierfür müssen die zur Verfügung stehenden Tasten mehrfach verwendet werden. Damit uns diese Mehrfach-Belegung leichter wird, finden wir beim Zubehör des Microtronic-Computers u. a. eine silberne Tastenfeld-Maske mit der Aufschrift (links unten) „Morse-Decoder“.

Die Tastenfeld-Maske bei geöffneter rauchglasfarbiger Abdeckhaube auf das Tastenfeld legen und Haube wieder schließen. In der oberen Reihe (zweite Taste von links) haben wir **SOS**. Bei Betätigung dieser Taste bringt der Computer automatisch das internationale Notrufzeichen „dreimal kurz – dreimal lang – dreimal kurz“. Die Taste, welche bisher nur für die Ziffer 0 verwendet wurde, ist jetzt auch für die Buchstaben A und B zuständig.

Betätigen wir diese Taste (in der untersten Reihe links), ergibt sich wie bisher fünfmal lang. Betätigen wir in der oberen Reihe die Taste mit der Bezeichnung **BU 1** und danach die Taste **OAB** ergibt sich „kurz-lang“, also das Morsezeichen **A**. Wird rechts oben die Taste **BU 2** und dann wieder die Taste **OAB** betätigt, ergibt sich „lang – dreimal kurz“, also das Morsezeichen **B**. Man gewöhnt sich schnell daran, daß für die Darstellung der Buchstaben jeweils zusätzlich eine der Tasten **BU 1** oder **BU 2** erforderlich ist.

Auch wenn wir nicht morsen können, sind wir nach kurzer Zeit in der Lage, komplette „Funksprüche“ zusammenzustellen. Mancher Amateur-Funker könnte vor Neid erblassen.

Die spezielle Funktion der Taste links oben mit **↑YZ** wird im folgenden Kapitel erklärt.

Die gesamte Versuchsanordnung und Programmierung bleibt für das folgende Kapitel bestehen.

Programm-Start wie üblich mit HALT – NEXT – 00 – RUN.

# Morse-Trainer

Das vorangegangene Programm „Morsezeichen-Decoder“ bietet uns die Möglichkeit, das Morse-Alphabet innerhalb kurzer Zeit zu erlernen. Auch wenn ein Lerneffekt nicht gewünscht wird, ist es interessant, diese Programm-Variation auszuprobieren.

Wir betätigen die Taste links oben **YZ**. Das Display zeigt: 0. Betätigen wir jetzt zweimal die Taste **OAB** morst der Computer automatisch die Morsezeichen A und B in unregelmäßiger Reihenfolge.

Wir brechen das Programm mit HALT ab und starten erneut mit NEXT – 00 – RUN. Erneut die Pfeiltaste betätigen, dann die Taste „**OAB**“ und danach „**1CD**“. Auf dem Display leuchten alle 6 Stellen und der Computer wiederholt den Morsevorgang der Buchstaben A und B sehr langsam, wodurch die Möglichkeit besteht, die Zeichen mitzuschreiben. Erkenntnis: Bei Eingabe von 00 (2 mal Taste „**OAB**“) werden die Zeichen schnell gemorst. Bei Eingabe (Taste „**OAB**“ und „**1CD**“) ergeben sich langsame Morsezeichen.

Wir brechen nochmals das Programm ab und starten neu mit HALT – NEXT – 00 – RUN, betätigen erneut die Pfeiltaste, dann „**1CD**“ und entweder nochmals „**1CD**“ (1 = langsam) oder „**OAB**“ (O = schnell) bringt der Computer automatisch die Morsezeichen A, B, C und D in unregelmäßiger Reihenfolge.

Wenn wir nach HALT – NEXT – 00 – RUN wieder die Pfeiltaste betätigen, dann „**2EF**“ (und wieder 0 für schnell oder 1 für langsam) morst der Computer die Buchstaben A, B, C, D und F in unregelmäßiger Reihenfolge.

Nach HALT – NEXT – 00 – RUN betätigen wir nun die Pfeiltaste zweimal nacheinander und zusätzlich 0 für langsam oder 1 für schnell. Der Computer bringt jetzt das gesamte Morse-Alphabet mit Buchstaben, Zeichen und Zahlen in unregelmäßiger, zufälliger Reihenfolge. Ein Morse-Automat für beliebig lange Funksprüche, deren Inhalt durch die zufällige Folge nicht immer sinnvoll sein wird. Zum Lernen des Morse-Alphabets jedoch ein excellentes Trainingsgerät.

Sollte der Computer die automatische Morsezeichen-Wiedergabe nach einiger Zeit selbstständig einstellen, ist er in einer „Endlos-Schleife“ angekommen, d. h. daß sich beim Programmieren ein Fehler eingeschlichen hat. Das gesamte Programm müßte nochmals genau kontrolliert werden.

## Das interessiert den Programmierer:

Um alle Buchstaben und Zahlen in Morsezeichen umzuwandeln, sind verschiedene „Programm-Tricks“ notwendig.

In der Programm-Tabelle haben wir auf den Adressen 00 bis 0D die „Eingabe-Routine“. Durch Abfrage, ob bzw. welche Buchstaben-Taste betätigt wurde, erfolgt ein Sprung zu den entsprechenden Programm-Teilen. Wurde keine Buchstaben-Taste betätigt, ist der eingegebene Wert eine Zahl (0 bis 9), die in Morsezeichen umgewandelt werden soll.

Die Umwandlung ist verhältnismäßig leicht lösbar. So bestehen z. B. alle Zahlen von 0 bis 9 aus fünf Morsezeichen (kurz oder lang, bzw. Punkt oder Strich). Die Zahlen 1 bis 5 beginnen jeweils mit Punkten und werden durch Striche fortgesetzt (z. B. 1 = „· – – –“, oder einmal kurz und viermal lang).

Bei den Zahlen 1 bis 5 wird in Register 1 (Eingabe-Register) die Anzahl der Punkte (kurze Töne) gespeichert. Im Register 5 wird die Anzahl der Striche (lange Töne) gespeichert. Die Anzahl der Striche (lange Töne) ist 5 minus Eingabe-Wert. Beispiel: Die Zahl 1. Fünf Striche minus Eingabe-Wert (Tasten-Wert) 1 ergibt vier Striche und ein Punkt. Durch den Sprung zum Unterprogramm „·“ wird ein kurzer Ton erzeugt und der in Register 1 enthaltene Wert um „1“ vermindert. Anschließend wird erneut ein kurzer Ton erzeugt, wodurch wiederum der Wert „1“ vom

Register-Inhalt 1 abgezogen wird. Dies wiederholt sich solange, bis der Inhalt der Registers 1 den Wert 0 erreicht hat. Danach wird im Unter-Programm „–“ ein langer Ton erzeugt und der Wert im Register 5 ebenfalls um „1“ vermindert. Auch dieser Vorgang wiederholt sich, bis der Inhalt des Registers 5 den Wert 0 erreicht hat. Wurde z. B. die Ziffer 3 zur Erzeugung des entsprechenden Morsezeichen eingegeben, ist das Register 1 für die drei Punkte (kurze Töne) und das Register 5 für die zwei Striche (lange Töne) zuständig.

Bei den Ziffern 6 bis 9 steht am Anfang des Morsezeichens jeweils ein Strich (langer Ton) gefolgt von einer entsprechenden Punktezahl (kurze Töne). In diesem Fall ist die Gesamtzahl der langen Töne der eingegebene Wert minus 5. Bei der Zahl 7 müssen z. B. zuerst zwei lange Töne und anschließend drei kurze Töne erzeugt werden.

Schwieriger wird die „Dekodierung“ von Buchstaben. Auch hierfür ist ein Trick anwendbar. Das längste Morsezeichen für Buchstaben besteht aus vier „Tönen“ (also Punkten oder Strichen).

Die Tonfolge kurz oder lang (Punkt oder Strich), kann durch eine Dual-Zahl erreicht werden. Beispiel: Das Morsezeichen F = „· · – –“

Verwenden wir für Punkte eine duale 0 und für Striche eine duale 1, dann ergibt sich für das Morsezeichen F die duale Darstellung 0010 was der Dual-Zahl 2 (siehe Tabelle Buchlesezeichen) entspricht.

Wird die duale Anordnung 0010 durch den SHL-Befehl nach links geschoben, ergibt sich beim ersten Schieben das Carry-Flag 0 und hierfür muß ein kurzer Ton (Punkt) erzeugt werden. Beim zweiten nach links Schieben ergibt das Carry-Flag wieder 0. Beim dritten Schieben ergibt sich ein Carry-Flag 1 (es muß ein langer Ton erzeugt werden) und beim vierten nach links Schieben wird das Carry-Flag wieder 0 (also kurzer Ton).

Der Buchstabe A hat das Morsezeichen: „· –“ und hierfür kann die duale Anordnung 0100 (Dual-Zahl 4) verwendet werden. Diese duale Anordnung darf jedoch nur zweimal nach links geschoben werden, weil ja für die Darstellung nur ein kurzer und ein langer Ton notwendig ist.

Wir müssen also die Buchstaben in einen zweistelligen Code verwandeln. Die erste Stelle des Codes gibt die Anzahl der Töne an, die zweite Code-Stelle die Dual-Zahl (die auch als „Bit-Muster“ bezeichnet wird). Das Bit-Muster (Dual-Zahl) wird für die Tonfolge (lang oder kurz) herangezogen. Der Buchstabe A hat somit den Code **24** (**2** = 2 Tonfolgen, **4** = Dual-Zahl 4 = 0100). Der Buchstabe F hat den Code **42** (**4** = 4 Tonfolgen, **2** = Dual-Zahl 2 = 0010).

Das Morsezeichen F dürfte einleuchtend sein. Die duale Anordnung 0010 ergibt das Morsezeichen: „· · – –“

Weniger einleuchtend ist die Methode beim Buchstaben A. Die duale Anordnung 0100 ergibt ein Morsezeichen „· – – –“. Ein A hat jedoch das Morsezeichen „· –“ (d. h., daß wir 2 Punkte zuviel haben).

Hier hilft der zweistellige Code, wobei die erste Stelle aussagt, wieviel Töne (kurz oder lang) erzeugt werden sollen. Beim Morsezeichen A werden somit nur die beiden ersten Bits (0 und 1) verwendet.

Die Umwandlung der Buchstaben in einen entsprechenden Code übernimmt das Programm ab Adresse 2B. Die Dual-Zahl (Bit-Muster) aus welcher sich die Tonfolge lang oder kurz ergibt, wird in Register 2 gespeichert. Die Anzahl der Tonfolgen (egal ob kurz oder lang) wird im Register 1 gespeichert. Die Tabelle „Bit-Muster“ zeigt uns, wie die Buchstaben des Alphabets in einen speziellen Code eingeteilt werden.

**Tabelle: Bit-Muster**

Buchstabe	Morsezeichen	Code in Reg. 1	Code in Reg. 2	Morsezeichen Länge (Anzahl Tonfolge)	„Bit-Muster“ (Verwendung der dualen Stellen)
A	•—	2	4	2	0 1 0 0
B	— • • •	4	8	1	1 0 0 0
C	— • — •	4	A	4	1 0 1 0
D	— • •	3	8	3	1 0 0 0
E	•	1	0	1	0 0 0 0
F	• • — •	4	2	4	0 0 1 0
G	— — •	3	C	3	1 1 0 0
H	• • • •	4	0	4	0 0 0 0
I	• •	2	0	2	0 0 0 0
J	• — — —	4	7	4	0 1 1 1
K	— • —	3	A	3	1 0 1 0
L	• — • •	4	4	4	0 1 0 0
M	— —	2	C	2	1 1 0 0
N	— •	2	8	2	1 0 0 0
O	— — —	3	E	3	1 1 1 0
P	• — — •	4	6	4	0 1 1 0
Q	— — • —	4	D	4	1 1 0 1
R	• — •	3	4	3	0 1 0 0
S	• • •	3	0	3	0 0 0 0
T	—	1	8	1	1 0 0 0
U	• • —	3	2	3	0 0 1 0
V	• • • —	4	1	4	0 0 0 1
W	• — —	3	6	3	0 1 1 0
X	— • • —	4	9	4	1 0 0 1
Y	— • — —	4	B	4	1 0 1 1
Z	— — • •	4	C	4	1 1 0 0

Ab Adresse 97 wird durch links Schieben (SHL-Befehl) der dualen Werte in Register 2 entweder ein langer oder ein kurzer Ton erzeugt. Die Anzahl der Töne wird durch das Herunterzählen (Minderung der Werte) in Register 1 festgelegt.

Der eigentliche Morseton wird in den Unter-Programmen (SUB-Routine) „.“ und „—“ erzeugt. Außerdem ergibt sich in diesen Unter-Programmen in Register A eine „Zufallszahl“, je nach dem, ob über den EINGANG ein Signal (von TAKT/CLOCK) vorhanden oder nicht vorhanden ist. Hierdurch wird die unregelmäßige Wiederholung der Morsezeichen A und B gesteuert.

Der Programm-Teil „Auto“ (automatische Morsezeichen-Erzeugung) beginnt bei Adresse BE. Die normalerweise erforderliche Tastatur-Eingabe wird durch die Erzeugung von Zufallszahlen ersetzt. Die Zufallszahl wird im Register D durch Hinzuzaddieren des Registers A erzeugt. Da in Register A der Wert relativ zufällig ist, ergibt sich in Register D eine ständig wechselnde Zufallszahl, wodurch das gesamte Morse-Alphabet in zufälliger Reihenfolge gemorst wird.

Bei diesem Programm wird der integrierte Zufalls-Generator nicht verwendet. Dies hat folgenden Grund: Die Erzeugung der Morsezeichen benötigt eine gewisse Zeit, in welcher der Zufalls-Generator eine bestimmte Schrittzahl weiterzählt. Diese Schrittzahl ist abhängig von der Morsezeichen-Länge. Würde man den integrierten Zufalls-Generator verwenden, ergäbe sich die ständige Wiederholung einer bestimmten Buchstaben-Folge. Durch die Verwendung eines speziell programmierter Zufalls-Generators, welcher durch den 1Hz-Takt gesteuert wird, ist eine solche Wiederholung nahezu ausgeschlossen.

Wir können dies ausprobieren, indem wir die Verbindungsleitung TAKT/CLOCK zum EINGANG 4 entfernen und den unter Adresse CB vorhandenen Befehl ADD A,D (4AD) durch den RND-Befehl (F05) ersetzen. Jetzt wird der Computer nur noch eine bestimmte Buchstaben-Folge morsen.

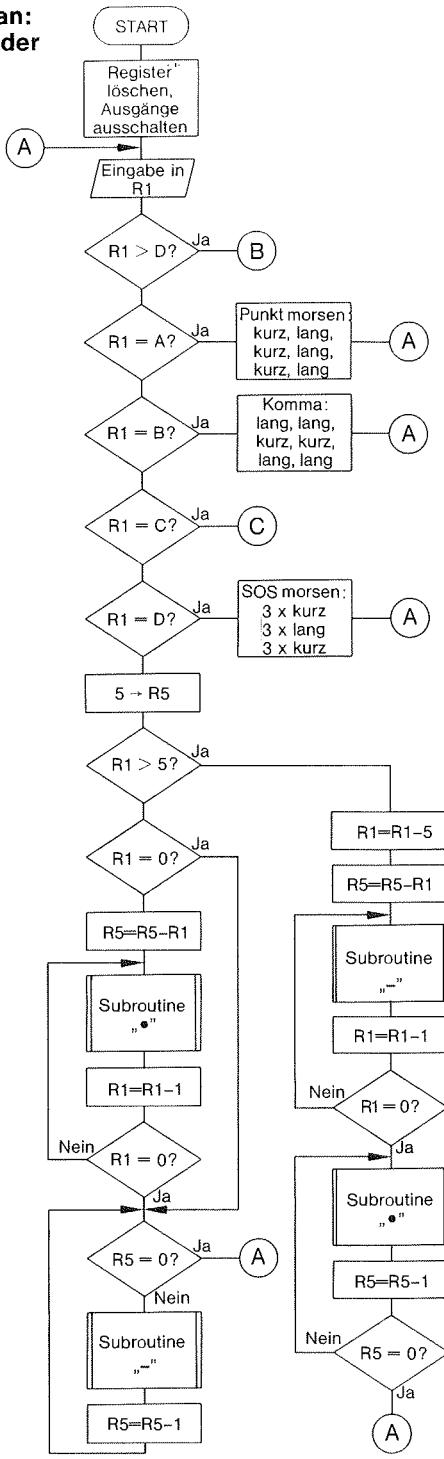
Warum wurde bei den Adressen 1D und 1E ein NOP-Befehl eingeschoben?

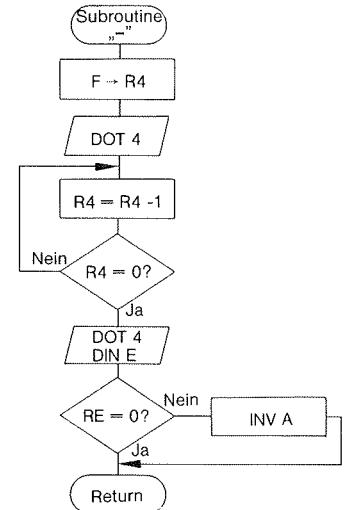
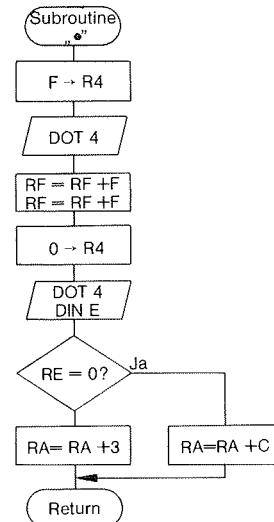
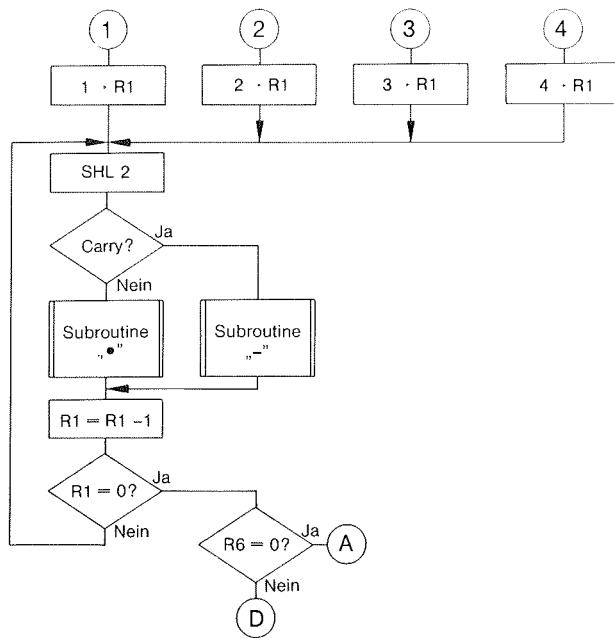
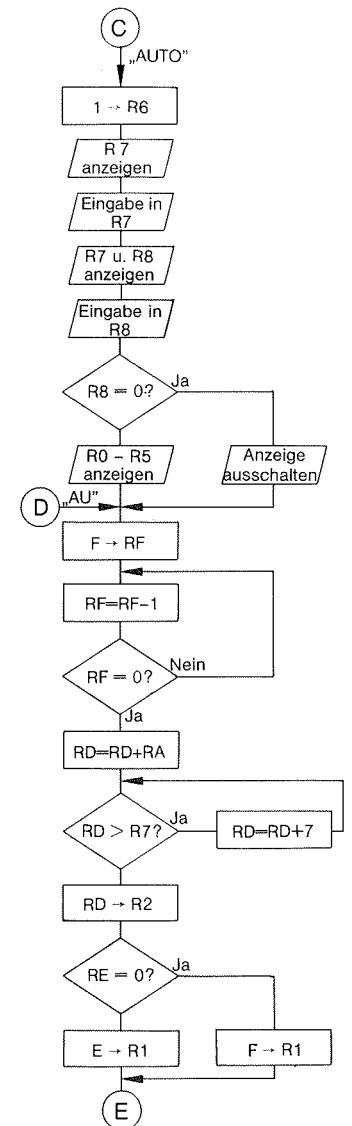
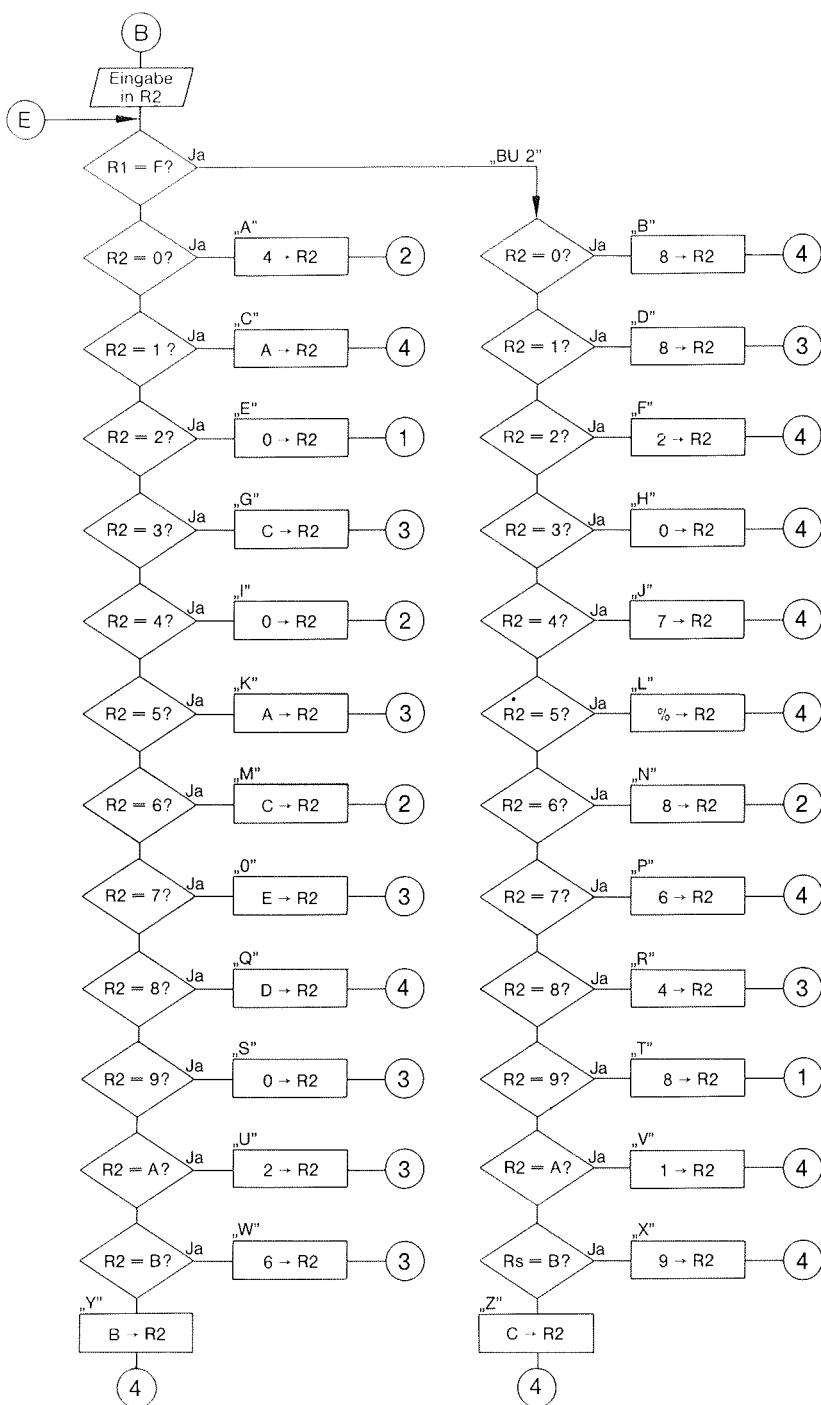
Es sollte lediglich demonstriert werden, daß es bei derart langen Programmen zweckmäßig ist, ab und zu einige NOP-Befehle einzugeben, weil sich hierdurch die Möglichkeit ergibt, einen bei der Eingabe evtl. vergessenen Befehl nachträglich noch dazwischenzuschieben, ohne daß das gesamte Programm neu eingegeben werden muß.

Vielleicht wundern wir uns auch über die Wiederholung gleicher Befehle? Dies ergibt sich z. B. bei den Programm-Stellen: „.“ und „—“. Hier erfolgt jedes Mal ein Sprung zum entsprechenden Unter-Programm, welches für die Erzeugung eines langen oder kurzen Tons verantwortlich ist. Ergeben sich z. B. für „SOS“ drei kurze – drei lange – drei kurze Töne, wird dreimal das bei Adresse D7 beginnende Unter-Programm „.“ nacheinander angesprungen, anschließend dreimal das bei Adresse E3 beginnende Unter-Programm „—“ usw.

Dieses Programm kann uns bei künftigen Programm-Überlegungen mancherlei Hinweise geben wie z. B. durch Verwendung der dualen Werte und Verringerung der dualen Stellen eine Vielzahl verschiedenartiger Signale erzeugt werden kann. Solche Signale können an die Ausgänge gebracht und für die unterschiedlichsten Möglichkeiten eingesetzt werden.

**Programm-Ablaufplan: Morsezeichen-Decoder (Morse-Trainer)**





## Widerstands-Code-Berechnung

Der Wert eines Widerstandes wird durch Farbringe gekennzeichnet. Für solche, die nicht ständig mit diesen Farb-Kennzeichnungen arbeiten, ist es oftmals schwierig, aus den Farbringen die richtigen Widerstandswerte zu ermitteln.

Das folgende Programm demonstriert, wie der Computer durch Eingabe verschiedener Farben die richtigen Widerstandswerte berechnen kann.

Dervom letzten Experiment noch angeschlossene Piezo-Summer wird nicht benötigt. Die evtl. ebenfalls noch vorhandene Tastenfeld-Maske entfernen und Programm nach HALT – NEXT – 00 eingeben.

### Programm: Widerstands-Code-Berechnung

Sprungziel	Adresse	Eingabe-Befehl	Mnemonic	Sprung nach	Erklärungen
	00	<b>F19</b>	DISP 1,9		Eingabe-Routine Taste: ↑
	01	<b>F08</b>	CLEAR		
	02	<b>FF0</b>	KIN 0		
	03	<b>9B0</b>	CMPI B,0		
	04	<b>D0D</b>	BRC 0D	START	
	05	<b>519</b>	ADDI 1,9		
	06	<b>949</b>	CMPI 4,9		
	07	<b>D00</b>	BRC 00		
	08	<b>034</b>	MOV 3,4		
	09	<b>023</b>	MOV 2,3		
	0A	<b>012</b>	MOV 1,2		
	0B	<b>001</b>	MOV 0,1		
	0C	<b>C02</b>	GOTO 02		
START	0D	<b>922</b>	CMPI 2,2		
	0E	<b>D14</b>	BRC 14	KILO	
	0F	<b>902</b>	CMPI 0,2		
	10	<b>E1E</b>	BRZ 1E	PRO	(PRO = Prozentuale Toleranz)
	11	<b>B34</b>	CALL 34	MORI	
	12	<b>712</b>	SUBI 1,2		
	13	<b>C0F</b>	,	GOTO 0F	Sprung Unterprogramm MORI
KILO	14	<b>952</b>	CMPI 5,2		
	15	<b>D1A</b>	BRC 1A	MEGA	
	16	<b>B34</b>	CALL 34	MORI	
	17	<b>1A3</b>	MOVI A,3		
	18	<b>732</b>	SUBI 3,2		
	19	<b>C0F</b>	GOTO 0F		
MEGA	1A	<b>B34</b>	CALL 34	MORI	
	1B	<b>1B3</b>	MOVI B,3		
	1C	<b>762</b>	SUBI 6,2		
	1D	<b>C0F</b>	GOTO 0F		
PRO	1E	<b>911</b>	CMPI 1,1		
	1F	<b>E2C</b>	BRZ 2C	1 → E	
	20	<b>921</b>	CMPI 2,1		
	21	<b>E2E</b>	BRZ 2E	2 → E	
	22	<b>9B1</b>	CMPI B,1		
	23	<b>E30</b>	BRZ 30	5 → E	
	24	<b>9A1</b>	CMPI A,1		
	25	<b>E32</b>	BRZ 32	1 → F	
	26	<b>12F</b>	MOVI 2,F		
ANZ	27	<b>F63</b>	DISP 6,3		
	28	<b>FF0</b>	KIN 0		
	29	<b>F2E</b>	DISP 2,E		

	2A	<b>FF0</b>	KIN 0		
	2B	<b>C00</b>	GOTO 00		
1 → E	2C	<b>11E</b>	MOVI 1,E		
	2D	<b>C27</b>	GOTO 27	ANZ	ANZ = Display anzeigen
2 → E	2E	<b>12E</b>	MOVI 2,E		
	2F	<b>C27</b>	GOTO 27	ANZ	
5 → E	30	<b>15E</b>	MOVI 5,E		
	31	<b>C27</b>	GOTO 27	ANZ	
1 → E	32	<b>11F</b>	MOVI 1,F		
	33	<b>C27</b>	GOTO 27		
MORI	34	<b>078</b>	MOV 7,8		
	35	<b>067</b>	MOV 6,7		
	36	<b>056</b>	MOV 5,6		
	37	<b>045</b>	MOV 4,5		
	38	<b>034</b>	MOV 3,4		
	39	<b>993</b>	CMPI 9,3		
	3A	<b>D3D</b>	BRC 3D		
	3B	<b>103</b>	MOVI 0,3		
	3C	<b>F07</b>	RET		
	3D	<b>104</b>	MOVI 0,4		
	3E	<b>F07</b>	RET		

Programm-Start: HALT – NEXT – 00 – RUN. Das Display zeigt: 0.

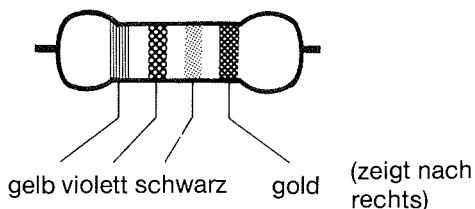
Die silberfarbige Tastenfeld-Maske mit der Bezeichnung „Widerstands-Code“ bei geöffneter Abdeckhaube auf das Tastenfeld legen und Haube wieder schließen.

Die meisten Widerstände haben 4 Farbringe, einer davon ist silber oder gold. Für die Widerstandswert-Berechnung muß der goldene, bzw. silberne Ring rechts sein. Die Eingabe beginnt also mit den Farben (siehe Abbildung): Gelb – violett – schwarz – gold. Das Display zählt die Tasten-Betätigungen und es wird jetzt die Ziffer 4 angezeigt. Die Farben-Eingabe wird durch Bestätigung der Taste ↑ (obere Tastenreihe) abgeschlossen.

Das Display zeigt 0047. Die eingegebenen Farbwerte entsprechen einem Widerstandswert von  $47\Omega$  (Ohm). Nochmalige Betätigung der ↑-Taste zeigt: 05, d. h. daß der berechnete Widerstandswert eine Toleranz von  $\pm 5\%$  hat.

Nochmalige Betätigung der ↑-Taste bringt auf dem Display wieder 0. Eine neue Eingabe kann vorgenommen werden.

## Farbwerte eines Widerstands



Auf der Computer-Platine sind eine ganze Menge Widerstände angeordnet, deren Werte auf diese Weise berechnet werden können. Wichtig ist es, daß grundsätzlich immer 4 Farben eingegeben werden. Bei Widerständen, die nur drei Farbringe haben, wird als vierte Farbe schwarz eingegeben.

Zeigt das Display z. B. nach der Eingabe von: Gelb – blau – violett – silber: 00460B, dann bedeutet die Bezeichnung B, daß es sich hier um einen Widerstand von  $460\text{ M}\Omega$  (Mega-Ohm) handelt. Ein Mega-Ohm hat eine Million Ohm.

Die Eingabe: Rot – orange – gelb – gold bringt das Ergebnis 230A. A bedeutet  $\text{K}\Omega$  (Kilo-Ohm), d. h. daß der berechnete Widerstandswert 230 k $\Omega$  hat. 1 Kilo-Ohm hat tausend Ohm.

### Das interessiert den Programmierer:

Die Funktion dieses Programms wird uns schnell verständlich, wenn wir nachstehende Widerstands-Farocode-Tabelle betrachten. Wir sehen, daß der erste und zweite Ring je nach Farbe einen bestimmten Wert darstellen. Der dritte Farbring zeigt an, um wieviele Nullen die Werte zu ergänzen.

## Widerstands-Farocode-Tabelle

Farbring 1			2			3			4 (Toleranz)		
Gold			5 %			10 %					
Silber											
Schwarz 0			0			-					
Braun			1			0					
Rot			2			00					
Orange			3			000					
Gelb			4			0000					
Grün			5			00000					
Blau			6			000000					
Violett			7			7					
Grau			8			8					
Weiß			9			9					

Würden wir beispielsweise dreimal die Farbe rot eingeben, ergibt sich 2200. Ein Widerstand mit den drei Farbringen rot hat somit 2200 $\Omega$ .

Geben wir z. B. ein: Gelb – violett – rot – gold, dann haben wir tatsächlich die Werte 4 – 7 – 2 – B eingegeben. (Für diese Feststellung wäre lediglich die aufgelegte Tastenfeld-Maske zu entfernen).

Das Display zeigt uns also die ersten beiden eingegebenen Ziffern direkt an. Für die dritte eingegebene Ziffer muß die Anzahl der Nullen ermittelt werden und lediglich für die Toleranz-Angabe B müssen die entsprechenden Werte ermittelt werden.

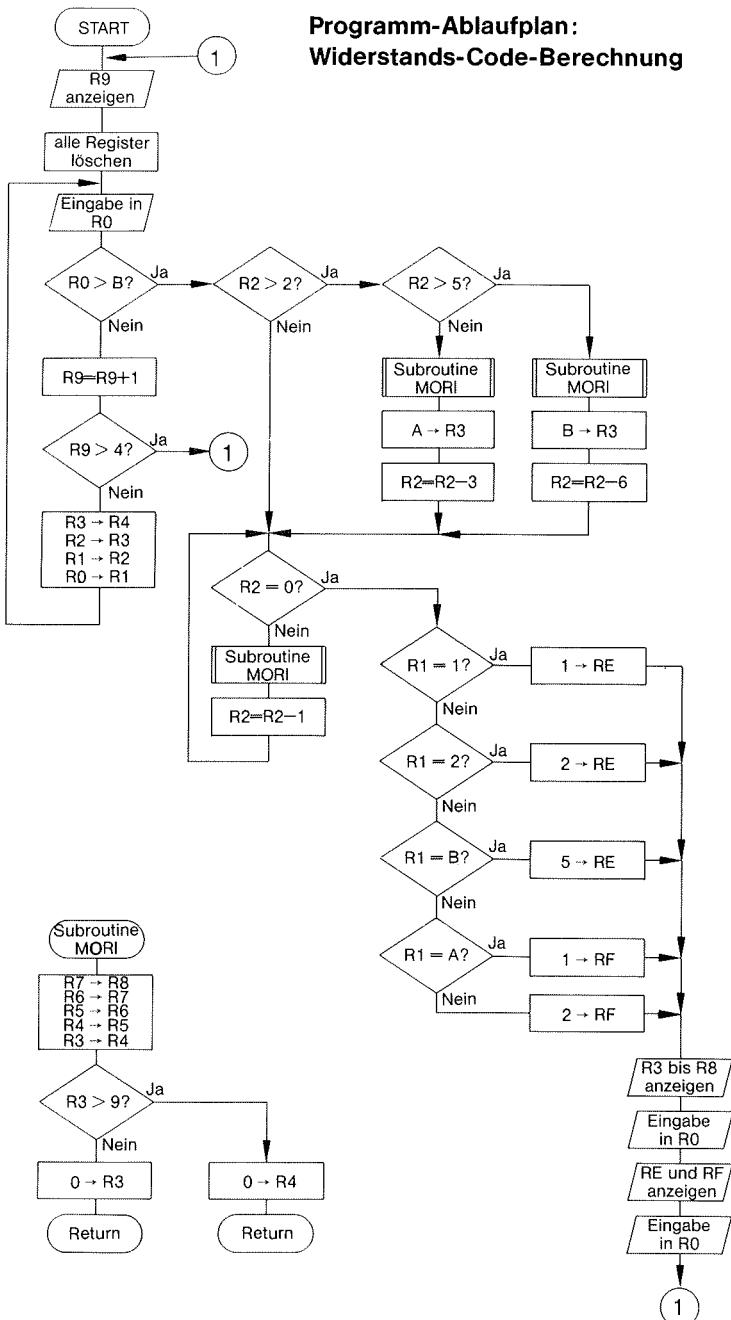
Die Eingabe erfolgt in die Register 1 bis 4. Die Register 3 und 4 werden direkt angezeigt. In Register 2 steht die Anzahl der anzugegenden Nullen. Somit müssen die angezeigten Werte der Register 3 und 4 im Display um die entsprechende Anzahl von Nullen nach links geschoben werden. Bei der Eingabe: Gelb – violett – rot ergibt sich sofort das richtige Ergebnis: 4700 $\Omega$ .

Damit die auf dem Display zur Verfügung stehenden 6 Anzeigestellen ausreichen, wird für drei Nullen ein A (Kilo) eingesetzt, bzw. für sechs Nullen ein B (Mega). Das angezeigte Ergebnis steht in den Registern 3 bis 8.

Die Toleranz wird durch Vergleichen festgestellt und in den Registern E und F gespeichert, bis sie durch erneute Pfeiltastenbetätigung angezeigt wird. Siehe auch Programm-Ablaufplan.

Das Programm-Beispiel „Widerstands-Code-Berechnung“ zeigt, wie durch logische Überlegungen mit einem sehr einfachen Programm ideale Problemlösungen erreicht werden können.

## Programm-Ablaufplan: Widerstands-Code-Berechnung



# Das „Taschen-Rechner“-Programm

Daß unser Computer addieren, subtrahieren, multiplizieren und dividiern kann, ist uns aus den Experimenten des Anleitungs-buches 1. Teil bekannt.

Diese Einzel-Programme sind nun zu einem komfortablen Rechner-Programm zusammengefaßt, mit welchem alle 4 Grundrechenarten ausgeführt werden können.

Das recht umfangreiche Programm sollte nach HALT – NEXT – 00 sehr sorgfältig eingegeben und kontrolliert werden.

## Programm: „Taschen-Rechner“

Sprung-ziel	Adr. Nr.	Eingabe-Befehl	Mne-monic	Sprung nach
	00	<b>F08</b>	CLEAR	
01	917	CMPI 1,7		
02	<b>E05</b>	BRZ 05		
03	<b>F0A</b>	RSC		
04	<b>C06</b>	GOTO 06		
05	<b>F09</b>	STC		
06	<b>F60</b>	OISP 6,0		
07	<b>FF6</b>	KIN 6		
08	<b>F02</b>	OISOUT		
09	<b>996</b>	CMPI 9,6		
0A	<b>D1D</b>	BRC 10		
0B	<b>947</b>	CMPI 4,7		
0C	<b>D15</b>	BRC 15		
0D	<b>187</b>	MOVI 8,7		
0E	<b>060</b>	MOV 6,0		
0F	<b>101</b>	MOVI 0,1		
10	<b>102</b>	MOVI 0,2		
11	<b>103</b>	MOVI 0,3		
12	<b>104</b>	MOVI 0,4		
13	<b>105</b>	MOVI 0,5		
14	<b>C03</b>	GOTO 03		
15	<b>045</b>	MOV 4,5		
16	<b>034</b>	MOV 3,4		
17	<b>023</b>	MOV 2,3		
18	<b>012</b>	MOV 1,2		
19	<b>001</b>	MOV 0,1		
1A	<b>060</b>	MOV 6,0		
1B	<b>387</b>	ANOI 8,7		
1C	<b>C03</b>	GOTO 03		
1D	<b>9A6</b>	CMPI A,6		
1E	<b>E33</b>	BRZ 33 „C/CE“		
1F	<b>317</b>	ANOI 1,7		
20	<b>90E</b>	CMPI 0,E		
21	<b>E2A</b>	BRZ 2A		
22	<b>9CE</b>	CMPI C,E		
23	<b>E66</b>	BRZ 66 +		
24	<b>9DE</b>	CMPI 0,E		
25	<b>E99</b>	BRZ 99 –		
26	<b>9EE</b>	CMPI E,E		
27	<b>E39</b>	BRZ 39 X		
28	<b>9FE</b>	CMPI F,E		
29	<b>E3F</b>	BRZ 3F :		
2A	<b>06E</b>	MOV 6,E		
2B	<b>07F</b>	MOV 7,F		

	2C	<b>008</b>	MOV 0,8	
	2D	<b>019</b>	MOV 1,9	
	2E	<b>02A</b>	MOV 2,A	
	2F	<b>03B</b>	MOV 3,B	
	30	<b>04C</b>	MOV 4,C	
	31	<b>05D</b>	MOV 5,D	
	32	<b>C01</b>	GOTO 01	
„C/CE“	33	<b>347</b>	ANOI 4,7	
	34	<b>947</b>	CMPI 4,7	
	35	<b>E00</b>	BRZ 00	
	36	<b>147</b>	MOVI 4,7	
	37	<b>106</b>	MOVI 0,6	
X	38	<b>C0E</b>	GOTO 0E	
	39	<b>06E</b>	MOV 6,E	
	3A	<b>B49</b>	CALL 49 „+/-“	
	3B	<b>B53</b>	CALL 53 „TAUSCH“	
	3C	<b>F0B</b>	MULT	
	3D	<b>D59</b>	BRC 59 EE	
	3E	<b>C61</b>	GOTO 61 =	
:	3F	<b>06E</b>	MOV 6,E	
	40	<b>B49</b>	CALL 49 „+/-“	
	41	<b>104</b>	MOVI 0,4	
	42	<b>105</b>	MOVI 0,5	
	43	<b>10C</b>	MOVI 0,C	
	44	<b>10D</b>	MOVI 0,D	
	45	<b>B53</b>	CALL 53 „TAUSCH“	
	46	<b>F0C</b>	DIV	
	47	<b>D59</b>	BRC 59 EE	
	48	<b>C61</b>	GOTO 61 EE	
„+/-“	49	<b>907</b>	CMPI 0,7	
	4A	<b>E4F</b>	BRZ 4F	
	4B	<b>90F</b>	CMPI 0,F	
	4C	<b>D51</b>	BRC 51	
	4D	<b>117</b>	MOVI 1,7	
	4E	<b>C52</b>	GOTO 52	
	4F	<b>90F</b>	CMPI 0,F	
	50	<b>D4D</b>	BRC 40	
	51	<b>107</b>	MOVI 0,7	
	52	<b>F07</b>	RET	
TAUSCH	53	<b>F0F</b>	EXRA	
	54	<b>06E</b>	MOV 6,E	
	55	<b>F0D</b>	EXRL	
	56	<b>F0F</b>	EXRA	
	57	<b>06E</b>	MOV 6,E	
	58	<b>F07</b>	RET	

EE	59	<b>1E0</b>	MOVI E,0	
	5A	<b>1E1</b>	MOVI E,1	
“	5B	<b>1E2</b>	MOVI E,2	
	5C	<b>1E3</b>	MOVI E,3	
	5D	<b>1E4</b>	MOVI E,4	
	5E	<b>1E5</b>	MOVI E,5	
	5F	<b>107</b>	MOVI 0,7	
	60	<b>C64</b>	GOTO 64	
“=“	61	<b>9BE</b>	CMPI B,E	
	62	<b>E64</b>	BRZ 64	
	63	<b>C2A</b>	GOTO 2A	
	64	<b>10E</b>	MOVI 0,E	
	65	<b>C01</b>	GOTO 01	
+	66	<b>06E</b>	MOV 6,E	
	67	<b>907</b>	CMPI 0,7	
	68	<b>E6D</b>	BRZ 60	
	69	<b>90F</b>	CMPI 0,F	
	6A	<b>EA2</b>	BRZ A2 „SU“	
	6B	<b>117</b>	MOVI 1,7	
	6C	<b>C71</b>	GOTO 71	
	6D	<b>90F</b>	CMPI 0,F	
	6E	<b>E70</b>	BRZ 70	
„AO“	6F	<b>CA2</b>	GOTO A2 „SU“	
	70	<b>107</b>	MOVI 0,7	
71	<b>480</b>	A0I 8,0	1. Reg.	
	72	<b>D76</b>	BRC 76	
	73	<b>990</b>	CMPI 9,0	
	74	<b>D76</b>	BRC 76	
	75	<b>C78</b>	GOTO 78	
	76	<b>560</b>	A0I 6,0	
	77	<b>511</b>	A0I 1,1	
	78	<b>491</b>	A0I 9,1	2. Reg.
	79	<b>D7D</b>	BRC 70	
	7A	<b>991</b>	CMPI 9,1	
	7B	<b>D7D</b>	BRC 7D	
	7C	<b>C7F</b>	GOTO 7F	
	7D	<b>561</b>	A0I 6,1	
	7E	<b>512</b>	A0I 1,2	
	7F	<b>4A2</b>	ADD A,2	3. Reg.
	80	<b>D84</b>	BRC 84	
	81	<b>992</b>	CMPI 9,2	
	82	<b>D84</b>	BRC 84	
	83	<b>C86</b>	GOTO 86	
	84	<b>562</b>	A0I 6,2	
	85	<b>513</b>	A0I 1,3	
	86	<b>4B3</b>	A0I B,3	4. Reg.
	87	<b>D8B</b>	BRC 8B	
	88	<b>993</b>	CMPI 9,3	
	89	<b>D8B</b>	BRC 8B	
	8A	<b>C8D</b>	GOTO 80	
	8B	<b>563</b>	A0I 6,3	
	8C	<b>514</b>	A0I 1,4	
	8D	<b>4C4</b>	ADD C,4	5. Reg.
	8E	<b>D92</b>	BRC 92	
	8F	<b>994</b>	CMPI 9,4	
	90	<b>D92</b>	BRC 92	

	91	<b>C94</b>	GOTO 94	
	92	<b>564</b>	ADDI 6,4	
	93	<b>515</b>	ADDI 1,5	
	94	<b>4D5</b>	ADD D,5	6. Reg.
	95	<b>D59</b>	BRC 59 „EE“	
	96	<b>995</b>	CMPI 9,5	
	97	<b>D59</b>	BRC 59 „EE“	
	98	<b>C61</b>	GOTO 61 „=“	
–	99	<b>06E</b>	MOV 6,E	
	9A	<b>907</b>	CMPI 0,7	
	9B	<b>E9F</b>	BRZ 9,F	
	9C	<b>90F</b>	CMPI 0,F	
	9D	<b>EA2</b>	BRZ A2	
	9E	<b>C6B</b>	GOTO 6B	
	9F	<b>90F</b>	CMPI 0,F	
	A0	<b>EA4</b>	BRZ A4	
A1	C6B	GOTO 6B		
„SU“	A2	<b>F0F</b>	EXRA	
A3	<b>06E</b>	MOV 6,E		
A4	<b>608</b>	SUB 0,8	Reg. 1	
A5	<b>DA7</b>	BRC A7		
A6	<b>CAA</b>	GOTO AA		
A7	<b>768</b>	SUBI 6,8		
A8	<b>719</b>	SUBI 1,9		
A9	<b>DAD</b>	BRC AD		
AA	<b>619</b>	SUB 1,9	Reg. 2	
AB	<b>DAE</b>	BRC AE		
AC	<b>CB1</b>	GOTO B1		
AD	<b>619</b>	SUB 1,9		
AE	<b>769</b>	SUBI 6,9		
AF	<b>71A</b>	SUBI 1,A		
BO	<b>DB4</b>	BRC B4		
B1	<b>62A</b>	SUB 2,A	Reg. 3	
B2	<b>DB5</b>	BRC B5		
B3	<b>CB8</b>	GOTO B8		
B4	<b>62A</b>	SUB 2,A		
B5	<b>76A</b>	SUBI 6,A		
B6	<b>71B</b>	SUBI 1,B		
B7	<b>DBB</b>	BRC BB		
B8	<b>63B</b>	SUB 3,B	4. Reg.	
B9	<b>DBC</b>	BRC BC		
BA	<b>CBF</b>	GOTO BF		
BB	<b>63B</b>	SUB 3,B		
BC	<b>76B</b>	SUBI 6,B		
BD	<b>71C</b>	SUBI 1,C		
BE	<b>DC2</b>	BRC C2		
BF	<b>64C</b>	SUB 4,C	5. Reg.	
CO	<b>DC3</b>	BRC C3		
C1	<b>CC6</b>	GOTO C6		
C2	<b>64C</b>	SUB 4,C		
C3	<b>76C</b>	SUBI 6,C		
C4	<b>71D</b>	SUBI 1,D		
C5	<b>DCC</b>	BRC CC		
C6	<b>65D</b>	SUB 5,0	6. Reg.	
C7	<b>DCD</b>	BRC CD		
C8	<b>F0F</b>	EXRA		

C9	<b>107</b>	MOVI 0,7	
CA	<b>06E</b>	MOV 6,E	
CB	<b>C61</b>	GOTO 61 „=“	
CC	<b>65D</b>	SUBI 5,0	
CD	<b>76D</b>	SUBI 6,0	
CE	<b>117</b>	MOVI 1,7	
CF	<b>F88</b>	INV 8	
D0	<b>768</b>	SUBI 6,8	
D1	<b>F89</b>	INV 9	
D2	<b>769</b>	SUBI 6,9	
D3	<b>F8A</b>	INV A	
D4	<b>76A</b>	SUBI 6,A	
D5	<b>F8B</b>	INV B	

D6	<b>76B</b>	SUBI 6,B	
D7	<b>F8C</b>	INV C	
D8	<b>76C</b>	SUBI 6,C	
D9	<b>F8D</b>	INV D	
DA	<b>76D</b>	SUBI 6,D	
DB	<b>110</b>	MOVI 1,0	
DC	<b>101</b>	MOVI 0,1	
DD	<b>102</b>	MOVI 0,2	
DE	<b>103</b>	MOVI 0,3	
DF	<b>104</b>	MOVI 0,4	
E0	<b>105</b>	MOVI 0,5	
E1	<b>C71</b>	GOTO 71 „AO“	

Ist ein Ergebnis größer als es die 6-stellige Display-Anzeige erlaubt, erscheint auf dem Display: EEEEEEE.

Für eine Division dürfen maximal 4 Stellen eingegeben werden. Divisions-Ergebnisse werden deshalb auch immer 4-stellig angezeigt. Da die Anzeige von Komma-Stellen nicht vorgesehen ist, werden Divisions-Ergebnisse grundsätzlich nach unten abgerundet: Beispiel:  $17 \div 2 = 8$  (anstelle des korrekten Ergebnisses 8,5).

### Das interessiert den Programmierer:

Zum besseren Verständnis soll zunächst die Register-Belegung in folgender Tabelle dargestellt werden:

### Register-Belegung:

Register Nr.	Register-Funktionen
0	Anzeige-Register
1	für Zahlen-Eingabe,
2	für Zwischen-Ergebnisse
3	und End-Ergebnisse
4	
5	
6	Eingabe-Register für Zahlen, die in die Reg. 0 bis 5 weitergeschoben werden
* 7	Plus-/Minus Index-Register (Speicherung, ob die in den Reg. 0 bis 5 vorhandenen Zahlenwerte positiv (+) oder negativ (-) sind.)
8	Register für die Zwischenspeicherung
9	der „weiter zu verarbeitenden“
A	Zahlenwerte, (Werte-Speicherung bis
B	Rechenoperation mit den in Reg. 0 bis 5 stehenden Zahlen ausgeführt wird).
C	
D	
E	Speicher-Register für die auszuführende Rechen-Operation (+ - x : )
* F	Plus-/Minus-Indexregister (Speicherung, ob die in den Reg. 8 bis D vorhandenen Zahlenwerte positiv (+) oder negativ (-) sind.)

\*) Die Plus-/Minus-Index-Register (Register 7 und F) haben den Inhalt 1 wenn die Zahlenwerte negativ sind, bzw. 0, wenn die Zahlenwerte positiv sind. Das Index-Register 7 erhält außerdem den Wert 4 sobald die C/CE-Taste einmal betätigt wurde, bzw. es erhält den Wert 8 sobald eine neue Zahl eingegeben wird.

Programm-Start mit HALT – NEXT – 00 – RUN. Das Display zeigt: 000000.

Bei geöffneter Abdeckhaube die Tastenfeld-Maske mit der Aufschrift „Taschen-Rechner/Calculator“ auf die Tastatur legen und Abdeckhaube wieder schließen.

Die Funktions-Tasten (auf der rechten Seite) werden für die Rechen-Operationen nicht benutzt. Dies betrifft vor allem die Taste C/CE, welche jetzt als „Löschen-Taste“ (in der zweiten Reihe von oben) vorhanden und zu benutzen ist.

Wie bei einem Taschen-Rechner werden die zu berechnenden Werte eingegeben. Die Eingabe der Zahlenwerte darf nicht zu schnell vorgenommen werden. Die Rechen-Operationen werden durch Betätigung der Tasten  $(+)$   $(-)$   $(\times)$   $(\div)$  ausgelöst und das Ergebnis wird durch die Taste  $(=)$  ermittelt und angezeigt. Mit der C/CE-Taste wird bei einmaliger Betätigung die zuletzt eingegebene Zahl gelöscht, bei zweimaliger Betätigung werden alle eingegebenen Zahlen gelöscht.

Ergibt sich ein negatives Ergebnis, leuchtet die obere Carry-Flag-LED gleichzeitig mit der Ergebnis-Anzeige auf.

Wir führen zunächst folgende Rechenaufgabe durch:  $25 \times 3 + 80 - 300$ . Die Eingaben, die zu betätigenden Tasten und die Display-Anzeige ergibt sich wie folgt:

Eingabe	Anzeige Display
25	0 0 0 0 2 5
x	0 0 0 0 2 5
3	0 0 0 0 0 3
+	0 0 0 0 7 5
80	0 0 0 0 8 0
-	0 0 0 1 5 5
300	0 0 0 3 0 0
=	0 0 0 1 4 5

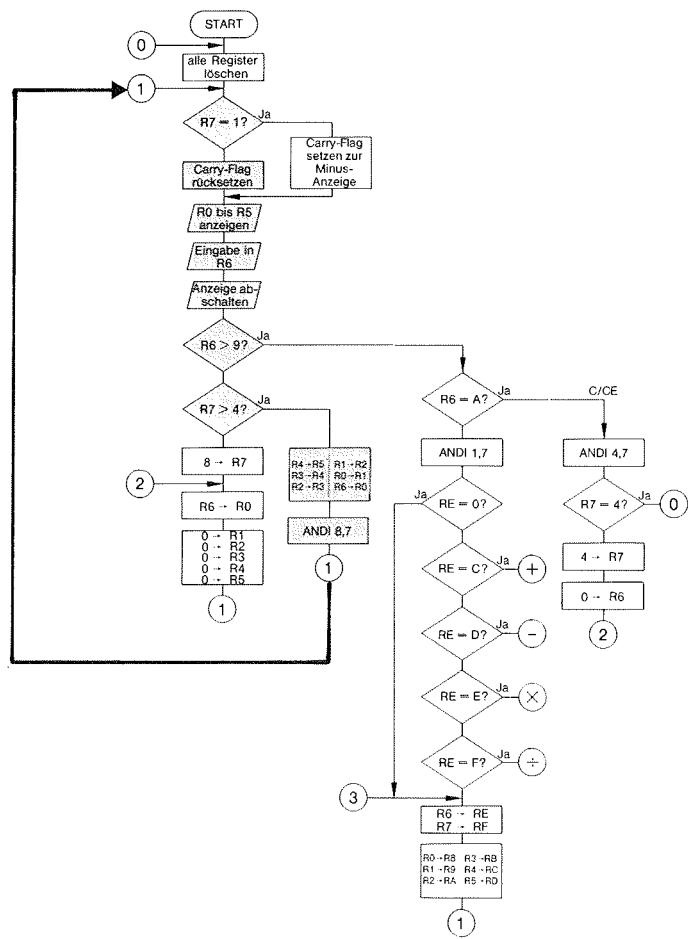
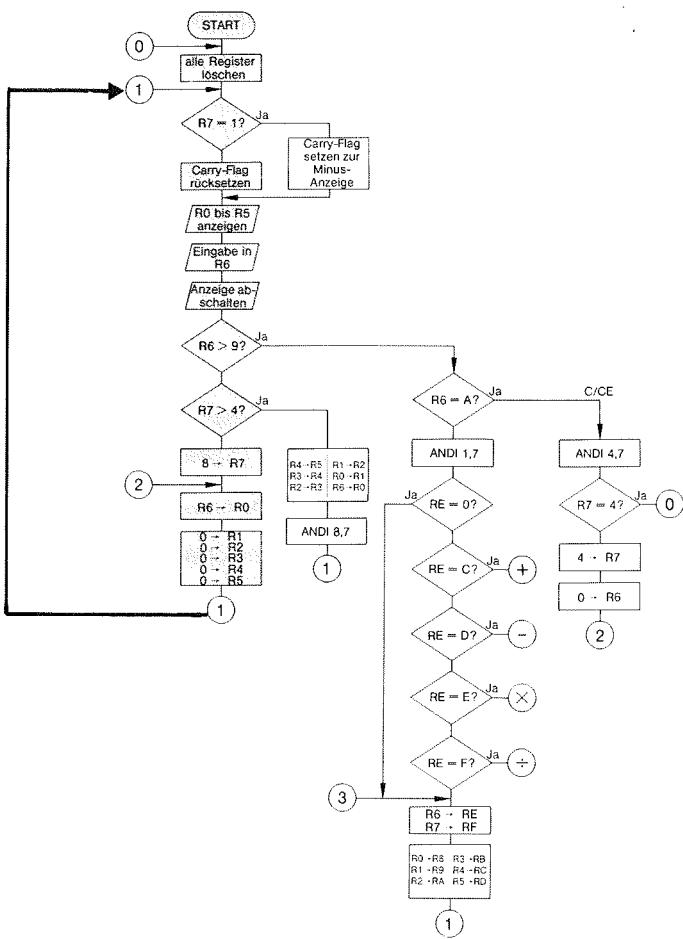
1. Zwischenergebnis

2. Zwischenergebnis

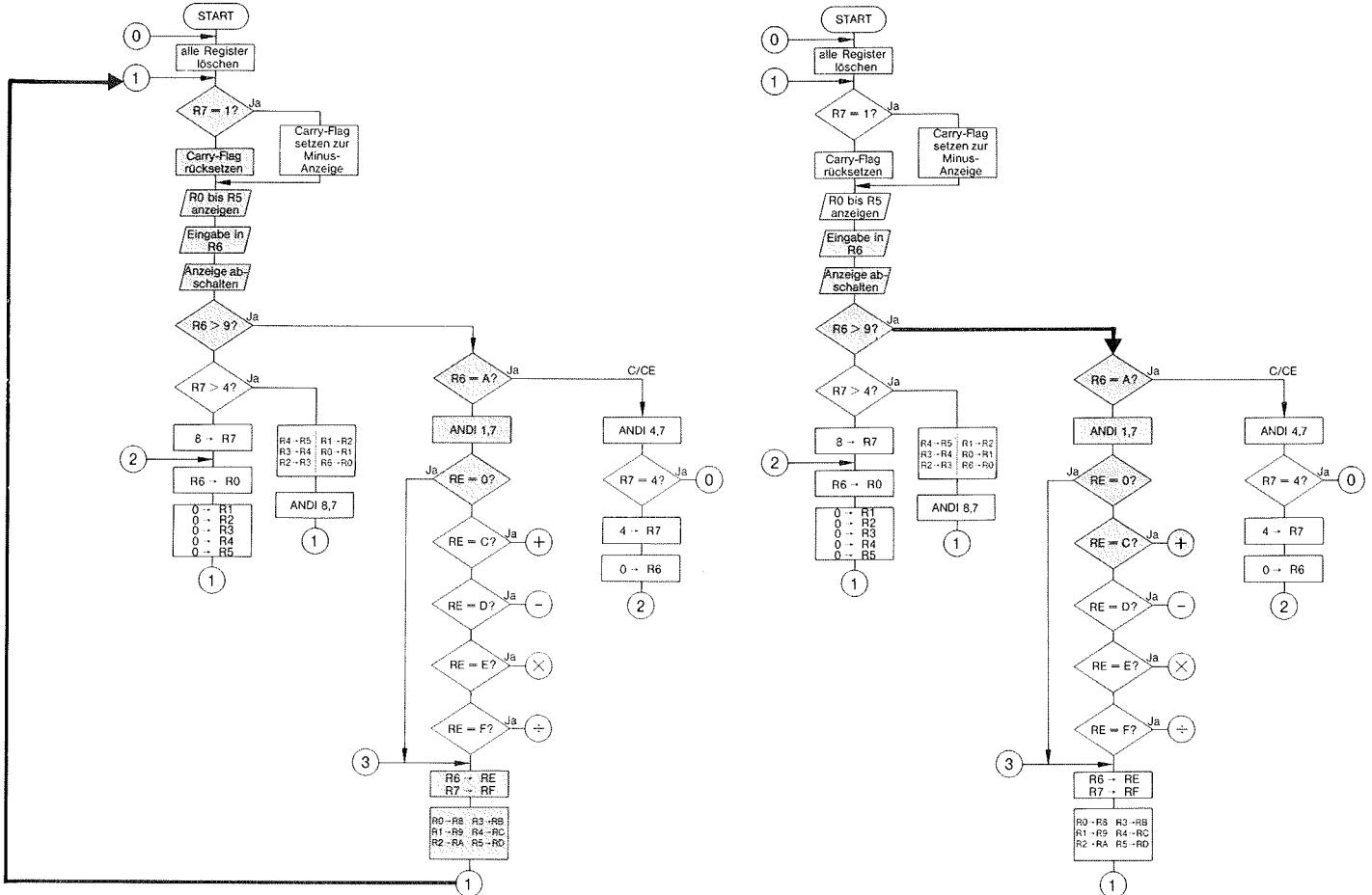
Endergebnis

Leuchtendes Carry-Flag = negatives Ergebnis  
d. h. Endergebnis = -145

Um die Programm-Funktion anschaulich zu demonstrieren, soll die Addition von zwei Zahlen in einem Teil des Programm-Ablaufplans dargestellt werden. Beispiel: Die Zahlen 24 und 32 sind zu addieren:



- Eingabe der ersten Zahl, z. B. der Zahlenwert 2. Nachdem die eingegebene Zahl nicht größer als 9 ist, und das Index-Register 7 den Wert 0 hat, wird der in Register 6 eingegebene Zahlenwert (2) in Register 0 gespeichert. Gleichzeitig werden die Register 1 bis 5 auf Null gesetzt. Das Index-Register 7 wird jetzt mit dem Wert 8 geladen.
  - Die zweite Zahleneingabe (z. B. 4) wird eingegeben. Nachdem das Index-Register 7 inzwischen den Wert 8 erhalten hat, erfolgt die Abfrage: „Ist der Inhalt von Register 7 größer als 4?“ – „Ja“ – somit erfolgt ein Sprung und die eingegebenen Zahlenwerte werden auf dem Display eine Stelle nach links geschoben.



3. Bevor die nächsten zu addierenden Zahlen eingegeben werden, wird die  $\oplus$ -Taste betätigt. Diese entspricht dem hexadezimalen Wert C, somit ist die jetzt in Register 6 vorgenommene Eingabe größer als 9. Durch den Vergleichs-Befehl (ist Inhalt Register 6 größer als 9?) erfolgt ein Sprung. Nachdem der Inhalt von Register 6 nicht den Wert A hat (siehe Vergleichs-Befehl) wird durch den Befehl ANDI 1,7 der Wert 8 im Index-Register Nr. 7 gelöscht. Das Register E hat momentan noch den Wert 0 und es erfolgt ein Sprung. Hierdurch wird der Inhalt von Register 6 in Register E gespeichert. Das Register E erhält nun durch den Wert C die Information, daß eine Addition vorgesehen ist (siehe auch Tabelle Register-Belegung). Die in den Registern 0 und 1 (oder falls mehrstellige Zahlen eingegeben werden bis Register 5) werden in den Registern 8 und 9 (oder bei mehrstelligeren Zahlen in den Registern 8 bis D) zwischengespeichert.

Jetzt werden die zu addierenden Zahlen (in unserem Beispiel 32) eingegeben, wodurch sich die bei 1. und 2. beschriebenen Vorgänge wiederholen.

4. Die Addition soll ausgeführt werden, die  $\ominus$ -Taste wird betätigt. Sie hat den Wert B, somit erfolgt ein Sprung wie bei 3. beschrieben. Nachdem in Register E der Wert C (durch die  $\oplus$ -Tasten-Eingabe) gespeichert wurde, ergibt sich jetzt ein Sprung zum Programm-Teil „AD“ (Addition), die im Gesamt-Programm-Ablaufplan durch den Kreis  $\oplus$  gekennzeichnet ist.

Am Anfang des Programm-Teils „Addition“ erfolgt ein Vergleich, ob eine der beiden zu addierenden Zahlen einen negativen Zahlenwert hat – falls „ja“, erfolgt ein Sprung zum Programm-Teil „SU“ (Subtraktion), im Programm-Ablaufplan durch den Kreis  $\ominus$  gekennzeichnet. Nachdem dies in unserem Beispiel nicht der Fall ist, erfolgt die dezimale Addition der einzelnen Register (Reg. 8 + Reg. 0, Reg. 9 + Reg. 1 usw.). Das Ergebnis wird angezeigt. Wäre das Ergebnis größer als die 6-stellige Anzeigemöglichkeit auf dem Display, würde ein Sprung zum Programm-Teil „EE“ erfolgen. Die Ergebnis-Anzeige wäre dann EEEEEEE.

Der Subtraktions-Teil beginnt ebenfalls mit einem Vergleich. Sind positive Zahlen zu subtrahieren (Beispiel:  $8 - 5 = 3$ ) werden die einzelnen Register subtrahiert und ggf. „dezimal korrigiert“, wie es im 1. Teil des Anleitungsbuches beschrieben wurde. Wird eine Subtraktion mit negativen Zahlen durchgeführt (Beispiel:  $-3 - 5 = -8$ ) dann ist zwar das Gesamtergebnis negativ, die beiden negativen Zahlenwerte müssen jedoch addiert werden. Bei einem sich ergebenden negativen Gesamtergebnis ergibt sich ein Carry-Übertrag, wodurch die einzelnen Stellen dezimal invertiert und anschließend der Wert 1 im Additions-Programm dazuaddiert werden, um ein korrektes Ergebnis zu erhalten (siehe Anleitungsbuch 1. Teil, Seite 51 bis 54).

Bei der Multiplikation  $\times$  und bei der Division  $\div$  wird zuerst im Unter-Programm „ $+/-$ “ festgestellt, ob das Ergebnis positiv oder negativ ist, damit der entsprechende Wert im Index-Register Nr. 7 gespeichert werden kann. Für Programmierer zeigt folgende Tabelle wie sich positive oder negative Ergebnisse errechnen lassen:

$$\begin{aligned} +1 \times +1 &= +1 \\ +1 \times -1 &= -1 \\ -1 \times +1 &= -1 \\ -1 \times -1 &= +1 \end{aligned}$$

Für ein positives Ergebnis wird im Index-Register 7 der Wert 0 gespeichert, für ein negatives Ergebnis der Wert 1. Anschließend werden die in den Registern 8 bis D vorhandenen Werte durch das Unter-Programm „Tausch“ in die Register 0 bis 5 gebracht. Jetzt wird die Multiplikation oder Division durchgeführt.

Alle 4 Berechnungen werden durch den Programm-Teil  $\equiv$  bzw. „EE“ abgeschlossen und es erfolgt ein Rücksprung in die Anzeige-Routine.

Die meisten Programm-Teile entsprechen den Ausführungen im 1. Teil des Anleitungsbuches. Neu ist jedoch, daß wir auch mit negativen Zahlen weiterrechnen können. Hierfür muß der Computer wissen, ob und welche Zahlen negativ sind.

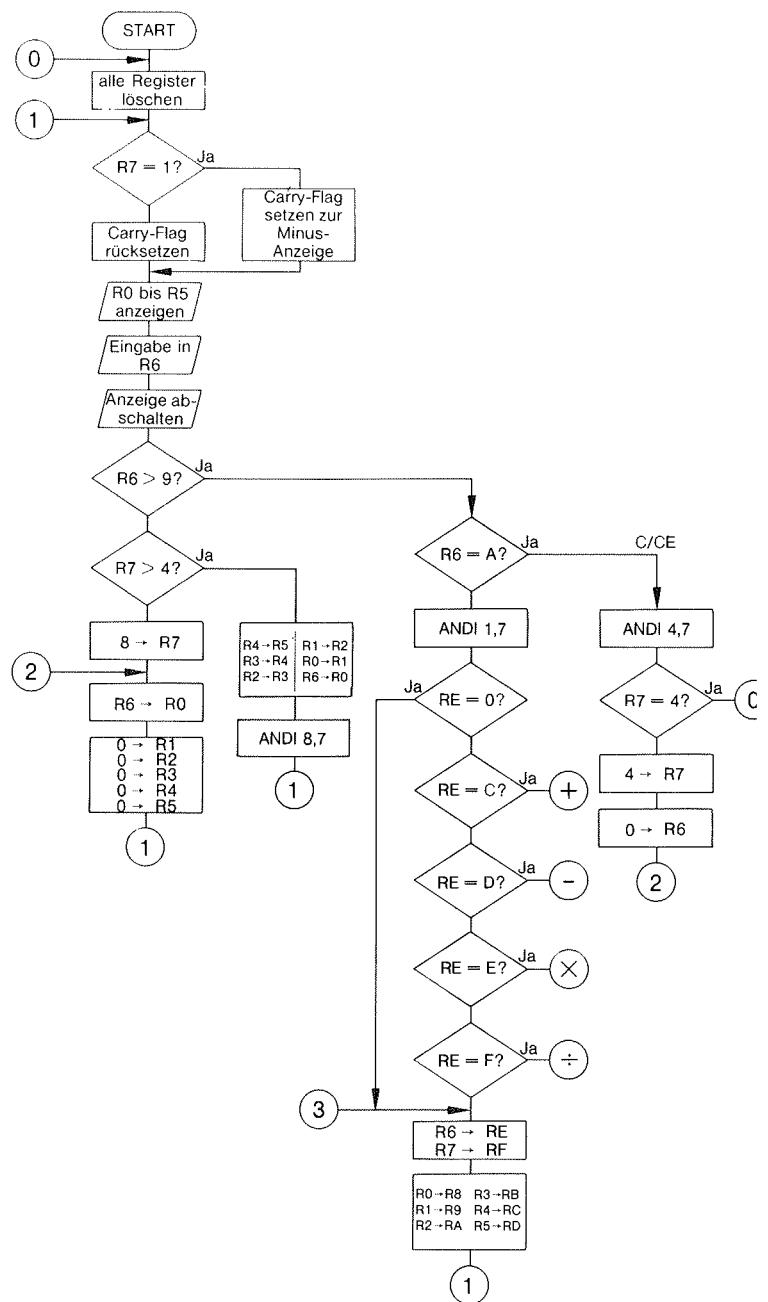
Hierfür stehen die Register 7 und F zur Verfügung.

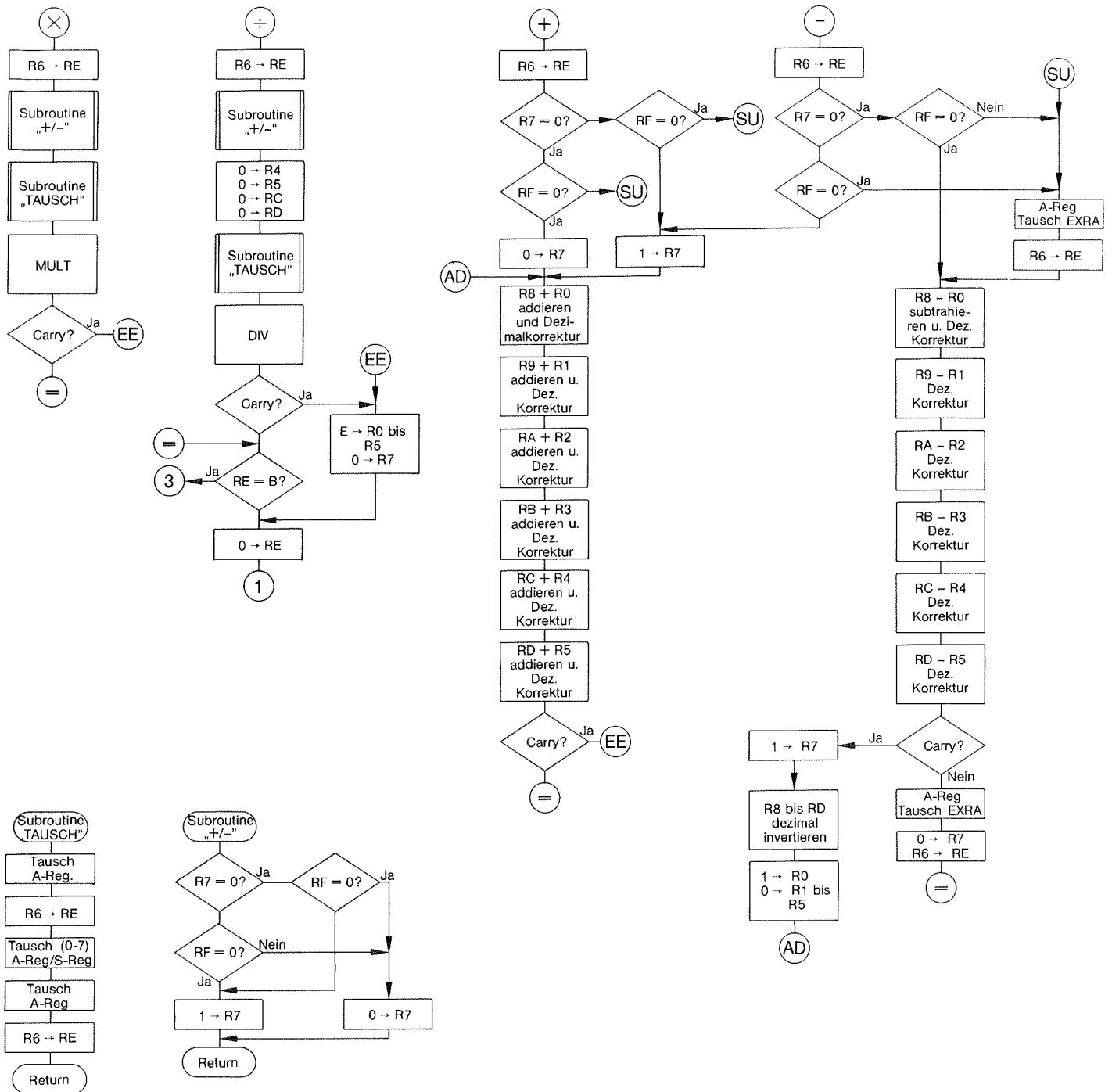
Beispiel: Durch eine Berechnung ergibt sich das negative Ergebnis  $-7$  und es soll hierzu der Wert  $+5$  hinzugezählt werden. Durch eine übliche Addition ist dies nicht zu erreichen, da  $7+5=12$  ergeben würde. Durch die Abfrage der Register 7 und F erfolgt ein Sprung zum Subtraktions-Programm „ $-$ “ und der Computer rechnet dann:  $5 - 7 = -2$ .

Ein ähnlicher Fall ergibt sich, wenn z. B. das letzte Ergebnis  $-7$  war und hiervon  $-5$  subtrahiert werden sollen. Durch Vergleichen findet der Computer heraus, daß beide Zahlen negativ sind, d. h. daß auch das Gesamt-Ergebnis eine negative Zahl ergeben muß. Er addiert deshalb beide Zahlen, wodurch sich das Ergebnis  $5 + 7 = 12$  ergibt. Da jedoch gespeichert war, daß die Addition mit negativen Zahlen ausgeführt werden muß, wird das Carry-Flag gesetzt, wodurch das negative Ergebnis:  $-12$  dargestellt wird.

Wenn man ein derartig komplexes Programm, wie z. B. dieses Rechner-Programm, selbst programmieren möchte, ist es erforderlich, jeden einzelnen Programm-Schritt zu durchdenken und geistig nachzuvollziehen.

### Gesamt-Programm-Ablaufplan: „Taschen-Rechner“





# Ein Programm speziell für Mathematiker: Sinus-Berechnung

Für mathematische Berechnungen werden häufig trigonometrische Funktionen (Winkelfunktionen) wie z. B. Sinus, Cosinus oder Tangens benötigt.

Es gibt technische/wissenschaftliche Taschen-Rechner, die solche Berechnungen durch Tastendruck lösen. Auch diese Taschen-Rechner enthalten meistens einen Mikroprozessor, welcher nur addieren und vielleicht gerade noch multiplizieren kann. Deshalb muß die Sinus-Berechnung aus Additionen, Multiplikationen usw. abgeleitet werden.

Für mathematisch Interessierte ist es sicherlich nicht uninteressant zu erfahren, wie z. B. ein Taschen-Rechner eine einfache Sinus-Berechnung ausführt.

Nach HALT – NEXT – 00 geben wir folgendes Programm ein:

## Programm: Sinus-Berechnung

Sprungziel	Adr. Nr.	Eingabe-Befehl	Mnemonic	Sprung nach
00	<b>F08</b>	CLEAR		
01	<b>F20</b>	DISP 2,0		
02	<b>FF1</b>	KIN 1		
03	<b>FF0</b>	KIN 0		
04	<b>F02</b>	DISOUT		
05	<b>F0D</b>	EXRL		
06	<b>F08</b>	CLEAR		
07	<b>140</b>	MOVI 4,0		
08	<b>171</b>	MOVI 7,1		
09	<b>112</b>	MOVI 1,2		
0A	<b>F0B</b>	MULT		
0B	<b>00C</b>	MOV 0,C		
0C	<b>01D</b>	MOV 1,0		
0D	<b>02E</b>	MOV 2,E		
0E	<b>03F</b>	MOV 3,F		
0F	<b>B59</b>	CALL 59	MOVI 0	
10	<b>0F1</b>	MOV F,1		
11	<b>0E0</b>	MOV E,0		
12	<b>F0D</b>	EXRL		
13	<b>0F1</b>	MDV F,1		
14	<b>0E0</b>	MOV E,0		
15	<b>F0B</b>	MULT		
16	<b>F0D</b>	EXRL		
17	<b>160</b>	MOVI 6,0		
18	<b>F0C</b>	DIV		
19	<b>F0D</b>	EXRL		
1A	<b>B59</b>	CALL 59	MDVI 0	
1B	<b>0F1</b>	MDV F,1		
1C	<b>0E0</b>	MOV E,0		
1D	<b>F0B</b>	MULT		
1E	<b>0C8</b>	MOV C,8		
1F	<b>0D9</b>	MDV D,9		
20	<b>0EA</b>	MOV E,A		
21	<b>0FB</b>	MOV F,B		
22	<b>628</b>	SUB 2,8		
23	<b>D25</b>	BRC 25		
24	<b>C28</b>	GOTO 28		
25	<b>768</b>	SUBI 6,8		
26	<b>719</b>	SUBI 1,9		
27	<b>D2B</b>	BRC 2B		
28	<b>639</b>	SUBI 3,9		

52	<b>D54</b>	BRC 54	
53	<b>C56</b>	GOTO 56	
54	<b>56A</b>	ADDI 6,A	
55	<b>51D</b>	ADDI 1,B	
56	<b>F48</b>	DISP 4,8	
57	<b>FF0</b>	KIN 0	
58	<b>C00</b>	GOTO 00	

MOVI 0	59	<b>100</b>	MOVI 0,0
5A	<b>101</b>	MOVI 0,1	
5B	<b>102</b>	MOVI 0,2	
5C	<b>103</b>	MOVI 0,3	
5D	<b>104</b>	MOVI 0,4	
5E	<b>105</b>	MDVI 0,5	
5F	<b>F07</b>	RET	

Programm-Start mit HALT – NEXT – 00 – RUN. Das Display zeigt: 00.

Der Sinus eines Winkels von 20° soll berechnet werden. Hierfür wird 20 eingegeben. Nach kurzer Berechnung zeigt das Display: 3415. Da unser Rechner ohne Kommastelle arbeitet, bedeutet dieses Ergebnis: 0,3415. Es ist zweckmäßig, für den Programm-Test die Beispiele einzugeben, damit das richtige Ergebnis kontrolliert werden kann. Bringt der Computer nicht das vorgenannte Ergebnis, muß die Programm-Eingabe kontrolliert und ggfls. korrigiert werden.

Vor Eingabe einer neuen Berechnung eine beliebige Zahlentaste betätigen, das Display zeigt wieder 00. Eine neue Berechnung kann vorgenommen werden. Da eine 2-stellige Eingabe erforderlich ist, muß für eine Sinus-Berechnung von 5° die zweistellige Eingabe 05 vorgenommen werden. In diesem Fall zeigt das Display: 0870, also 0,0870.

Wir sollten noch beachten, daß der eingegebene Winkel nicht größer als 45° sein darf, weil die über 45° hinausgehenden Berechnungen nicht mehr korrekt vorgenommen werden.

### Das interessiert den Programmierer:

Bevor wir uns mit dem eigentlichen Programm-Ablauf beschäftigen, müssen wir wissen, wie durch Additionen, Multiplikationen usw. eine Sinus-Funktion berechnet werden kann.

Die Winkelfunktionen lassen sich mit Hilfe sogenannter „Reihen“ berechnen. Die Reihe zur Berechnung eines Sinus hat folgende Formel:

$$\sin x = X - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \dots \dots \dots$$

Ein zu berechnender Sinus wird umso genauer, je mehr Glieder die zu berechnende Reihe aufweist. Damit das Programm nicht zu aufwendig wird, wurde die Reihe nach dem dritten Glied abgebrochen (umrandete Formel).

Für ein Beispiel wollen wir den Sinus des Winkels von 20° ausrechnen. Wir können in diese Formel jedoch nicht „20“ für „x“ direkt einsetzen. Zunächst muß der Winkel in das Bogenmaß umgerechnet werden, indem wir „20“ mit der Zahl „π“ (3,1415) multiplizieren und anschließend durch die Zahl 180 dividieren:

$$20^\circ = \frac{20 \times 3,1415}{180} = 0,349$$

20° entsprechen einem Bogenmaß 0,349

0,349 kann jetzt in die „Reihe“ eingesetzt werden. Um auch Nicht-Mathematikern das Nachvollziehen dieser Berechnungen zu erleichtern, werden die Berechnungen in nachfolgender Darstellung Schritt für Schritt vollzogen:

$$\sin 20^\circ = 0,349 - \frac{0,349^3}{3!} + \frac{0,349^5}{5!}$$

$$\sin 20^\circ = 0,349 - \frac{0,349 \times 0,349 \times 0,349}{1 \times 2 \times 3} + \frac{0,249 \times 0,349 \times 0,349 \times 0,349}{1 \times 2 \times 3 \times 4 \times 5}$$

$$\sin 20^\circ = 0,349 - \frac{0,0425}{6} + \frac{0,00518}{120}$$

$$\sin 20^\circ = 0,349 - 0,0071 + 0,0000432$$

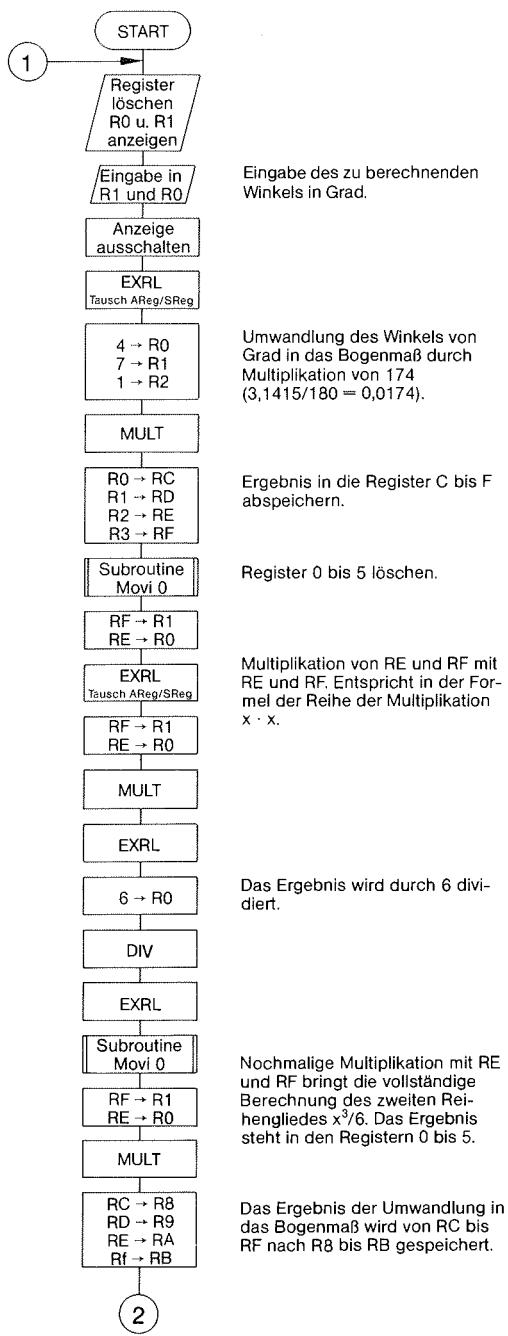
$$\sin 20^\circ = 0,3419$$

Unsere Berechnung bringt das Ergebnis 0,3419 während der Computer 0,3415 gerechnet hat. Diese geringfügige Abweichung ergibt sich durch Auf- und Abrundungen des Computers. Da sich jedoch maximal eine Abweichung von 0,3% ergibt, ist die Genauigkeit der Berechnungen unseres Computers für die meisten Anwendungsfälle ausreichend.

Aus den vorgenannten Berechnungs-Beispielen ist es ersichtlich, daß viele Multiplikationen durchzuführen sind. Da das Ergebnis dieser Multiplikationen immer kleiner als 1 ist, müßte mit Kommastellen gerechnet werden. Hierfür ist jedoch der Mikroprozessor nicht vorgesehen. Durch auf- und abgerundete Multiplikationen wird mit ganzen Zahlen gerechnet, wodurch eine Menge Programmierarbeit gespart wird.

Die Winkelwerte werden in die Register 0 und 1 eingegeben und anschließend durch Multiplikation mit 174 in das Bogenmaß umgewandelt ( $3,1415 : 180 = 0,0174$ ).

### Programm-Ablaufplan: Sinus-Berechnung



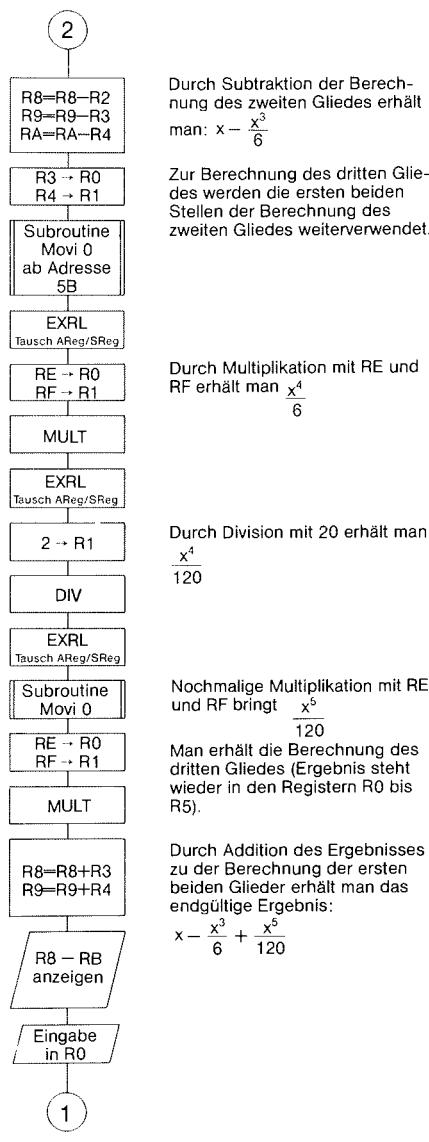
Das Ergebnis wird in den Registern C bis F gespeichert. Die beiden ersten Ziffern (Register E und F) werden multipliziert, anschließend durch 6 dividiert und nochmals mit den Registern E und F multipliziert. Hierdurch erhalten wir das zweite Glied in der Reihe:  $x^3 \div 6$ .

Anschließend werden die Register C bis F (eingegebener Wert im Bogemaß) in die Register 8 bis B umgespeichert. Dann wird das Ergebnis der letzten Berechnung abgezogen. Anschließend wird das Ergebnis der Berechnung des zweiten Gliedes ( $x^3 \div 6$ ) ebenfalls mit den Registern E und F multipliziert, durch die Zahl 20 dividiert und abermals mit den Registern E und F multipliziert. Hierdurch erhalten wir das dritte Glied der Reihe ( $x^5 \div 120$ ).

Dieses Ergebnis wird mit dem in den Registern 8 bis B stehenden Ergebnis addiert, das Gesamtergebnis steht jetzt in diesen Registern und wird angezeigt.

Es ist noch beachtenswert, daß nicht immer mit allen Ziffern eines Ergebnisses weitergerechnet wird, sondern das teilweise nur die ersten zwei oder drei Ziffern weiter berechnet werden.

Der Programm-Ablaufplan enthält einige weitere Erklärungen, für die prinzipielle Programm-Funktion.



# Cosinus- und Tangens-Berechnungen

Unser Programm Sinus-Berechnung kann auch für andere Winkelfunktions-Berechnung „umgebaut“ werden. Hierzu folgende Anregungen:

Die COSINUS-Funktion läßt sich ebenfalls mit Hilfe einer „Reihe“ berechnen:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!}$$

Es ist zu beachten, daß für „x“ der Winkel in das Bogenmaß umgerechnet werden muß. Die meisten Taschen-Rechner können intern nur die Winkel von  $0^\circ$  bis  $45^\circ$  eines Sinus oder Cosinus berechnen. Soll z. B. der Sinus von  $60^\circ$  ermittelt werden, berechnet der Taschen-Rechner den Cosinus von  $90^\circ$  minus  $60^\circ$ , also von  $30^\circ$ . Dies ist möglich, weil folgende Beziehung besteht:

$$\sin \delta = \cos (90^\circ - \delta)$$

$$\text{bzw. } \cos \delta = \sin (90^\circ - \delta)$$

Durch diesen einfachen Trick kann einige Rechenarbeit eingespart werden.

Die TANGENS-Funktion kann ermittelt werden indem der Sinus und Cosinus berechnet wird und anschließend die beiden Ergebnisse miteinander dividiert werden:

$$\tan \delta = \sin \delta : \cos \delta$$

Durch derartige Überlegungen und Manipulationen kann und muß der Mikroprozessor enorme Rechenleistungen ausführen.

Ein dreiteiliges Super-Programm!

1. Teil:

## Berechnung der Tage zwischen zwei Daten

Kein Problem ist es, die Tage auszurechnen, die sich beispielsweise zwischen dem 3. und dem 17. eines Monats ergeben. Schwieriger wird es, wenn wir wissen möchten, wieviel Tage zwischen dem 5. März 1981 und dem 24. Dezember 1985 liegen.

Für die Berechnung dieser Tage hilft uns das folgende Programm, wobei evtl. Schaltjahre automatisch berücksichtigt werden. Die Berechnungen basieren auf dem Gregorianischen Kalender.

Bevor wir mit dem Programmieren beginnen, sollte der gesamte Programm-Speicher gelöscht werden, weil die „Tages-Berechnung“ der erste Teil eines Super-Programms darstellt, welches durch die beiden folgenden Kapitel ergänzt wird. Es ist auch zu beachten, daß wir nach der Adresse 5C den zweiten Programm-Teil bei Adresse 88 fortsetzen. Die beiden Programm-Teile werden später durch das Bio-Rhythmus-Programm vervollständigt.

Löschen des Programm-Speichers: HALT – PGM – 5.

Nach HALT – NEXT – 00 sehr sorgfältig die Programmierung vornehmen:

### Programm: Berechnung der Tage zwischen 2 Daten

Sprungziel	Adr. Nr.	Eingabe-Befehl	Mnemonic	Sprung nach
	00	<b>F08</b>	CLEAR	
	01	<b>F0E</b>	EXRM	
	02	<b>F08</b>	CLEAR	
	03	<b>F6C</b>	DISP 6,C	
	04	<b>FF2</b>	KIN 2	
	05	<b>992</b>	CMPI 9,2	
	06	<b>D0E</b>	BRC DE	
	07	<b>001</b>	MOV 0,1	
	08	<b>0F0</b>	MOV F,0	
	09	<b>0EF</b>	MOV E,F	
	0A	<b>0DE</b>	MOV O,E	
	0B	<b>0CD</b>	MOV C,D	
	0C	<b>02C</b>	MOV 2,C	
	0D	<b>C03</b>	GOTO 03	
	0E	<b>F02</b>	DISDUT	
	0F	<b>017</b>	MOV 1,7	
	10	<b>006</b>	MOV 0,6	
	11	<b>B9E</b>	CALL 9E	MOVI 0
	12	<b>0C0</b>	MOV C,0	
	13	<b>0D1</b>	MOV 0,1	
	14	<b>BAE</b>	CALL AE	SUBI 1
	15	<b>F76</b>	MAS 6	
	16	<b>F77</b>	MAS 7	
	17	<b>F0D</b>	EXRL	
	18	<b>B9E</b>	CALL 9E	MOVI 0
	19	<b>110</b>	MOVI 1,0	
	1A	<b>131</b>	MOVI 3,1	
	1B	<b>F0B</b>	MULT	
	1C	<b>BBC</b>	CALL BC	ADD
	1D	<b>90D</b>	CMPI 0,0	
	1E	<b>D2D</b>	BRC 20	MO
	1F	<b>92C</b>	CMPI 2,C	
	20	<b>D2D</b>	BRC 20	MD
	21	<b>B99</b>	CALL 99	MOV J
	22	<b>BAE</b>	CALL AE	SUBI 1

23	<b>F0D</b>	EXRL	
24	<b>140</b>	MOVI 4,0	
25	<b>F0C</b>	OIV	
26	<b>BBC</b>	CALL BC	ADD
27	<b>B99</b>	CALL 99	MOV J
28	<b>BAE</b>	CALL AE	SUBI 1
29	<b>10D</b>	MOVI 0,0	
2A	<b>C88</b>	GOTO 88	INT
2B	<b>BDF</b>	CALL DF	SUB
2C	<b>C43</b>	GOTD 43	BERE
MD	<b>B9E</b>	CALL 9E	MOVI 0
2E	<b>0D1</b>	MOV D,1	
2F	<b>0C0</b>	MOV C,0	
30	<b>F0D</b>	EXRL	
31	<b>140</b>	MOVI 4,0	
32	<b>F0B</b>	MULT	
33	<b>134</b>	MDVI 3,4	
34	<b>125</b>	MOVI 2,5	
35	<b>BA5</b>	CALL A5	ADD 2
36	<b>010</b>	MOV 1,0	
37	<b>101</b>	MDVI 0,1	
38	<b>BDF</b>	CALL DF	SUB
39	<b>B99</b>	CALL 99	MOV J
3A	<b>F0D</b>	EXRL	
3B	<b>140</b>	MDVI 4,0	
3C	<b>F0C</b>	DIV	
3D	<b>BBC</b>	CALL BC	ADD
3E	<b>B99</b>	CALL 99	MOV J
3F	<b>11D</b>	MOVI 1,0	
40	<b>C88</b>	GOTO 88	INT
41	<b>10D</b>	MOVI 0,0	
42	<b>BDF</b>	CALL DF	SUB
BERE	<b>B99</b>	CALL 99	MOV J
44	<b>F0D</b>	EXRL	
45	<b>B9E</b>	CALL AE	MOVI 0
46	<b>150</b>	MOVI 5,0	

	47	<b>161</b>	MOVI 6,1	
	48	<b>132</b>	MOVI 3,2	
	49	<b>F0B</b>	MULT	
	4A	<b>BBC</b>	CALL BC ADD	
	4B	<b>B9E</b>	CALL 9E MOVI 0	
	4C	<b>F20</b>	OISP 2,0	
	4D	<b>FF1</b>	KIN 1	
	4E	<b>FF0</b>	KIN 0	
	4F	<b>F02</b>	DISDUT	
	50	<b>BBC</b>	CALL BC ADD	
	51	<b>F0E</b>	EXRM	
	52	<b>90F</b>	CMPI 0,F	
	53	<b>D57</b>	BRC 57	
	54	<b>1FF</b>	MOVI F,F	
	55	<b>F7F</b>	MAS F	
	56	<b>C02</b>	GOTD 02	
	57	<b>F0F</b>	EXRA	
	58	<b>F0E</b>	EXRM	
	59	<b>BDF</b>	CALL DF SUB	
	5A	<b>F58</b>	DISP 5,8	
	5B	<b>FF0</b>	KIN 0	
	5C	<b>C00</b>	GOTD 00	
		<b>„HALT-NEXT-88“.</b>		
		Dann weiter programmieren =		
INT	88	<b>10C</b>	MDVI 0,C	
	89	<b>F0D</b>	EXRL	
	8A	<b>B9E</b>	CALL 9E MOVI 0	
	8B	<b>112</b>	MDVI 1,2	
	8C	<b>F0C</b>	DIV	
	8D	<b>114</b>	MOVI 1,4	
	8E	<b>BA5</b>	CALL A5 ADD 2	
	8F	<b>F0D</b>	EXRL	
	90	<b>B9E</b>	CALL 9E MDVI 0	
	91	<b>130</b>	MOVI 3,0	
	92	<b>F0B</b>	MULT	
	93	<b>F0D</b>	EXRL	
	94	<b>140</b>	MDV 4,0	
	95	<b>F0C</b>	DIV	
	96	<b>90D</b>	CMP 0,D	
	97	<b>E2B</b>	BRZ 2B	
	98	<b>C41</b>	GOTD 41	
MDV J	99	<b>0E0</b>	MOV E,0	
	9A	<b>0F1</b>	MDV F,1	
	9B	<b>062</b>	MDV 6,2	
	9C	<b>073</b>	MDV 7,3	
	9D	<b>CA2</b>	GOTO A2	
MDVI 0	9E	<b>100</b>	MDVI 0,0	
	9F	<b>101</b>	MOVI 0,1	
	A0	<b>102</b>	MOVI 0,2	
	A1	<b>103</b>	MDVI 0,3	
	A2	<b>104</b>	MDVI 0,4	
	A3	<b>105</b>	MDVI 0,5	
	A4	<b>F07</b>	RET	
ADD 2	A5	<b>440</b>	ADD 4,0	
	A6	<b>DAA</b>	BRC AA	
	A7	<b>990</b>	CMPI 9,0	

	A8	<b>DAA</b>	BRC AA	
	A9	<b>CAC</b>	GOTO AC	
	AA	<b>560</b>	ADDI 6,0	
	AB	<b>511</b>	ADDI 1,1	
	AC	<b>451</b>	ADD 5,1	
	AD	<b>CA2</b>	GOTO A2	
SUBI 1	AE	<b>710</b>	SUBI 1,0	
	AF	<b>DB1</b>	BRC B1	
	B0	<b>F07</b>	RET	
	B1	<b>760</b>	SUBI 6,0	
	B2	<b>711</b>	SUBI 1,1	
	B3	<b>DB5</b>	BRC B5	
	B4	<b>F07</b>	RET	
	B5	<b>761</b>	SUBI 6,1	
	B6	<b>712</b>	SUBI 1,2	
	B7	<b>DB9</b>	BRC B9	
	B8	<b>F07</b>	RET	
	B9	<b>762</b>	SUBI 6,2	
	BA	<b>713</b>	SUBI 1,3	
	BB	<b>F07</b>	RET	
ADD	BC	<b>408</b>	ADD 0,8	
	BD	<b>DC1</b>	BRC C1	
	BE	<b>998</b>	CMPI 9,8	
	BF	<b>DC1</b>	BRC C1	
	CO	<b>CC3</b>	GOTD C3	
	C1	<b>568</b>	ADDI 6,8	
	C2	<b>519</b>	ADDI 1,9	
	C3	<b>419</b>	ADD 1,9	
	C4	<b>DC8</b>	BRC C8	
	C5	<b>999</b>	CMPI 9,9	
	C6	<b>DC8</b>	BRC C8	
	C7	<b>CCA</b>	GOTD CA	
	C8	<b>569</b>	ADDI 6,9	
	C9	<b>51A</b>	ADDI 1,A	
	CA	<b>42A</b>	ADD 2,A	
	CB	<b>DCF</b>	BRC CF	
	CC	<b>99A</b>	CMPI 9,A	
	CD	<b>DCF</b>	BRC CF	
	CE	<b>CD1</b>	GOTO D1	
	CF	<b>56A</b>	ADDI 6,A	
	D0	<b>51B</b>	ADDI 1,B	
	D1	<b>43B</b>	ADD 3,B	
	D2	<b>DD6</b>	BRC D6	
	D3	<b>99B</b>	CMPI 9,B	
	D4	<b>DD6</b>	BRC D6	
	D5	<b>CD8</b>	GOTO D8	
	D6	<b>56B</b>	ADDI 6,B	
	D7	<b>51C</b>	ADDI 1,C	
	D8	<b>44C</b>	ADD 4,C	
	D9	<b>DDD</b>	BRC DD	
	DA	<b>99C</b>	CMPI 9,C	
	DB	<b>DDD</b>	BRC DD	
	DC	<b>F07</b>	RET	
	DD	<b>56C</b>	ADDI 6,C	
	DE	<b>F07</b>	RET	

SUB	DF	<b>608</b>	SUB 0,8	
	E0	<b>DE2</b>	BRC E2	
	E1	<b>CE5</b>	GOTO E5	
	E2	<b>768</b>	SUBI 6,8	
	E3	<b>719</b>	SUBI 1,9	
	E4	<b>DE8</b>	BRC E8	
	E5	<b>619</b>	SUB 1,9	
	E6	<b>DE9</b>	BRC E9	
	E7	<b>CEC</b>	GOTO EC	
	E8	<b>619</b>	SUB 1,9	
	E9	<b>769</b>	SUBI 6,9	
	EA	<b>71A</b>	SUBI 1,A	
	EB	<b>DEF</b>	BRC EF	
	EC	<b>62A</b>	SUB 2,A	
	ED	<b>DF0</b>	BRC F0	
	EE	<b>CF3</b>	GOTO F3	
	EF	<b>62A</b>	SUB 2,A	

F0	<b>76A</b>	SUBI 6,A	
F1	<b>71B</b>	SUBI 1,B	
F2	<b>DF6</b>	BRC F6	
F3	<b>63B</b>	SUB 3,B	
F4	<b>DF7</b>	BRC F7	
F5	<b>CFA</b>	GOTO FA	
F6	<b>63B</b>	SUB 3,B	
F7	<b>76B</b>	SUBI 6,B	
F8	<b>71C</b>	SUBI 1,C	
F9	<b>DFD</b>	BRC FD	
FA	<b>64C</b>	SUB 4,C	
FB	<b>DFE</b>	BRC FE	
FC	<b>F07</b>	RET	
FD	<b>64C</b>	SUB 4,C	
FE	<b>76C</b>	SUBI 6,C	
FF	<b>F07</b>	RET	

Programm-Start mit HALT – NEXT – 00 – RUN. Das Display zeigt: 000000.

Zur Berechnung, wieviele Tage sich zwischen dem 5. 3. 1981 und dem 24. 12. 1985 ergeben, müssen wir zunächst die „ältere“ Jahreszahl eingeben also: 1981. Dann den Monat 03. Das Display zeigt: 198103.

Taste A betätigen – nach einigen Sekunden zeigt das Display: 00. Jetzt wird noch der Tag eingegeben: 05. Das Display zeigt jetzt wieder: 000000. Es erfolgt die zweite Eingabe, zunächst das Jahr (1985), dann den Monat (12). Das Display zeigt: 198512. Nach Betätigung der Taste A das Tagesdatum eingeben (24).

Der Computer nimmt die Berechnung vor. Wenn die Daten und das Programm korrekt eingegeben wurden, zeigt das Display die zwischen den beiden Daten liegenden Tage: 1755.

Nach Betätigung einer beliebigen Zahlentaste zeigt das Display wieder: 000000 und es können die Tage zwischen zwei anderen Daten berechnet werden.

Es ist wichtig, daß immer zuerst die älteren (kleineren) Daten eingegeben werden. Für Monate und Tage sind immer 2-stellige Eingaben notwendig, deshalb sind für kleinere Eingaben als 10 eine 0 voranzustellen (für Juli = 7. Monat – Eingabe 07). Die Datum-Eingabe darf nicht zu schnell vorgenommen werden, und auf dem Display sollte die richtige Eingabe kontrolliert werden.

Beispiel: Bei einer Eingabe zwischen den Jahren 1. 1. 1800 (180001 – 01) und dem 1. 1. 2200 (220001 – 01) bringt der Computer das Ergebnis: 46097. Tatsächlich liegen jedoch 146097 Tage zwischen den beiden Daten.

Das eingegebene Programm wird auch für die beiden noch folgenden Kapitel benötigt, deshalb nicht löschen oder verändern.

# Berechnung eines Wochentages innerhalb eines beliebigen Jahres

Mit diesem Programm können wir durch Eingabe eines Datums berechnen, welcher Wochentag (Sonntag, Montag usw.) zu diesem Zeitpunkt gewesen war.

Das bereits eingegebene Programm muß ab Adresse 51 bei einigen Programm-Schritten geändert werden. Wir nehmen die Eingabe nach HALT – NEXT – 51 vor, indem die neuen Befehle anstelle der bisher vorhandenen Befehle eingegeben werden (durch Eingabe eines neuen Befehls-Codes wird der bisher eingegebene Code gelöscht):

## Programm-Änderung: Berechnung des Wochentages

Nachdem wir bei Adresse 6D den letzten Befehl eingegeben haben, wird das Programm wieder gestartet: HALT – NEXT – 00 – RUN.

Das Display zeigt wieder: 000000.

Wir möchten zunächst ein leicht überprüfbares Datum testen:

Auf welchen Wochentag fällt der Heilige Abend im Jahr 1982?

Wie beim vorangegangenen Programm wird das Datum 24.12.1982 wieder wie folgt eingegeben: Jahreszahl und Monat (198212) – Taste A betätigen – dann Wochentag (24). Das Display zeigt 6. Aus der Wochentag-Tabelle sehen wir, daß die Zahl 6 ein Freitag ist.

Das Programm ist so ausgelegt, daß Wochentag-Berechnungen ab dem 1.1.1918 bis zum 31.12.2189 vorgenommen werden können. Liegen die eingegebenen Daten vor oder nach diesen Jahreszahlen, sind die angezeigten Ergebnisse falsch.

## Wochentag-Tabelle

0 = Samstag  
1 = Sonntag  
2 = Montag  
3 = Dienstag  
4 = Mittwoch  
5 = Donnerstag  
6 = Freitag

Das Programm wird für das folgende Kapitel noch benötigt, bitte nicht löschen oder verändern.

## Programm-Ergänzung: Wochentag-Berechnung

Sprungziel	Adr. Nr.	Eingabe-Befehl	Mnemonic	Sprung nach
	51	<b>B9E</b>	CALL 9E	MOVI 0
	52	<b>090</b>	MOV 9,0	
	53	<b>0A1</b>	MOV A,1	
	54	<b>0B2</b>	MOV B,2	
	55	<b>0C3</b>	MOV C,3	
	56	<b>F0D</b>	EXRL	
	57	<b>170</b>	MOVI 7,0	
	58	<b>F0C</b>	DIV	
	59	<b>006</b>	MOV 0,6	
	5A	<b>017</b>	MOV 1,7	
	5B	<b>02E</b>	MOV 2,E	
	5C	<b>F0D</b>	EXRL	
	5D	<b>001</b>	MOV 0,1	
	5E	<b>080</b>	MOV 8,0	
	5F	<b>F0D</b>	EXRL	
	60	<b>B9E</b>	CALL 9E	MOVI 0
	61	<b>170</b>	MOVI 7,0	
	62	<b>F0C</b>	DIV	
	63	<b>061</b>	MOV 6,1	
	64	<b>072</b>	MOV 7,2	
	65	<b>0E3</b>	MOV E,3	
	66	<b>F0D</b>	EXRL	
	67	<b>B9E</b>	CALL 9E	MOVI 0
	68	<b>170</b>	MOVI 7,0	
	69	<b>F0B</b>	MULT	
	6A	<b>BDF</b>	CALL DF	
	6B	<b>F18</b>	DISP 1,8	
	6C	<b>FF0</b>	KIN 0	
	6D	<b>C00</b>	GOTO 00	

# Berechnung des Bio-Rhythmus

Die Theorie des Bio-Rhythmus besagt, daß in jedem Menschen 3 verschiedene, sich ständig wiederholende Zyklen ablaufen, welche am Tag der Geburt beginnen.

Der erste Zyklus, auch als physischer oder körperlicher Rhythmus bezeichnet, hat eine Länge von 23 Tagen. Der zweite Zyklus, psychischer oder seelischer Rhythmus, hat eine Länge von 28 Tagen. Der dritte Zyklus, ein geistiger Rhythmus, hat eine Länge von 33 Tagen.

Nachdem der Bio-Rhythmus davon ausgeht, daß das körperliche, seelische und geistige Leben des Menschen in einem wellenförmigen und nach Tagen genau abgrenzbaren Höhen und Tiefen verläuft, können aus dem Bio-Rhythmus die „besseren“ und „schlechteren“ Tage abgeleitet werden.

Befinden wir uns in der ersten Hälfte des Bio-Rhythmus, sind wir in einer guten Phase. In der zweiten Rhythmus-Hälfte soll angeblich unsere geistige und körperliche Konstitution etwas schlechter sein.

Für einen Sportler könnte in erster Linie der körperliche Rhythmus interessant sein, während für einen geistig arbeitenden Menschen auch der geistige Rhythmus eine wesentlichere Rolle spielen kann. Ganz allgemein gilt, daß sich der Mensch umso besser fühlt, je mehr Rhythmen sich in der ersten Zyklus-Hälfte befinden.

Inwieweit die umstrittene Lehre des Bio-Rhythmus (von W. Fließ und H. R. Früh) verlässlich anwendbar ist, soll dahingestellt bleiben. Für den Computer ergeben sich durch die Ermittlungen des persönlichen Bio-Rhythmus sehr manigfaltige Rechen-Operationen.

Unsere bio-rhythmische Konstitution können wir mit folgendem Programm ermitteln.

Das bereits eingegebene Programm „Berechnung der Tage zwischen zwei Daten“ bleibt bestehen. Es wird jedoch ab Adresse 51 bis zu Adresse 87 geändert, bzw. ergänzt.

Wir beginnen die Änderungs-Programmierung nach HALT – NEXT – 51:

## Ergänzungs-Programm: Bio-Rhythmus

Durch Eingabe dieses Programm-Teils ist unser Programm-Speicher von Adresse 00 bis zur letztmöglichen Adresse FF „gefüllt“.

Sprungziel	Adr. Nr.	Eingabe-Befehl	Mnemonic	Sprung nach
	51	<b>F0E</b>	EXRM	
	52	<b>90F</b>	CMPI 0,F	
	53	<b>D57</b>	BRC 57	
	54	<b>1FF</b>	MOVI F,F	
	55	<b>F7F</b>	MAS F	
	56	<b>C02</b>	GOTO 02	
	57	<b>F0F</b>	EXRA	
	58	<b>F0E</b>	EXRM	
	59	<b>BDF</b>	CALL DF	SUB
	5A	<b>136</b>	MOVI 3,6	
	5B	<b>127</b>	MOVI 2,7	
	5C	<b>11F</b>	MOVI 1,F	
	5D	<b>C65</b>	GOTO 65	
	5E	<b>186</b>	MOVI 8,6	
	5F	<b>127</b>	MOVI 2,7	
	60	<b>12F</b>	MOVI 2,F	
	61	<b>C65</b>	GOTO 65	

	62	<b>136</b>	MOVI 3,6	
	63	<b>137</b>	MOVI 3,7	
	64	<b>13F</b>	MOVI 3,F	
	65	<b>F76</b>	MAS 6	
	66	<b>F77</b>	MAS 7	
	67	<b>B9E</b>	CALL 9E	MOVI 0
	68	<b>090</b>	MOV 9,0	
	69	<b>0A1</b>	MOV A,1	
	6A	<b>0B2</b>	MOV B,2	
	6B	<b>0C3</b>	MOV C,3	
	6C	<b>F0D</b>	EXRL	
	6D	<b>060</b>	MOV 6,0	
	6E	<b>071</b>	MOV 7,1	
	6F	<b>F0C</b>	DIV	
	70	<b>F0D</b>	EXRL	
	71	<b>012</b>	MOV 1,2	
	72	<b>001</b>	MOV 0,1	
	73	<b>080</b>	MOV 8,0	
	74	<b>F0D</b>	EXRL	
	75	<b>B9E</b>	CALL 9E	MOVI 0
	76	<b>060</b>	MOV 6,0	
	77	<b>071</b>	MOV 7,1	
	78	<b>F0C</b>	DIV	
	79	<b>F0D</b>	EXRL	
	7A	<b>012</b>	MOV 1,2	
	7B	<b>001</b>	MOV 0,1	
	7C	<b>100</b>	MOVI 0,0	
	7D	<b>F0D</b>	EXRL	
	7E	<b>B9E</b>	CALL 9E	MOVI 0
	7F	<b>060</b>	MOV 6,0	
	80	<b>071</b>	MOV 7,1	
	81	<b>F0C</b>	DIV	
	82	<b>F10</b>	DISP 1,0	
	83	<b>FF0</b>	KIN 0	
	84	<b>92F</b>	CMPI 2,F	
	85	<b>E62</b>	BRZ 62	
	86	<b>D00</b>	BRC 00	
	87	<b>C5E</b>	GOTO 5E	

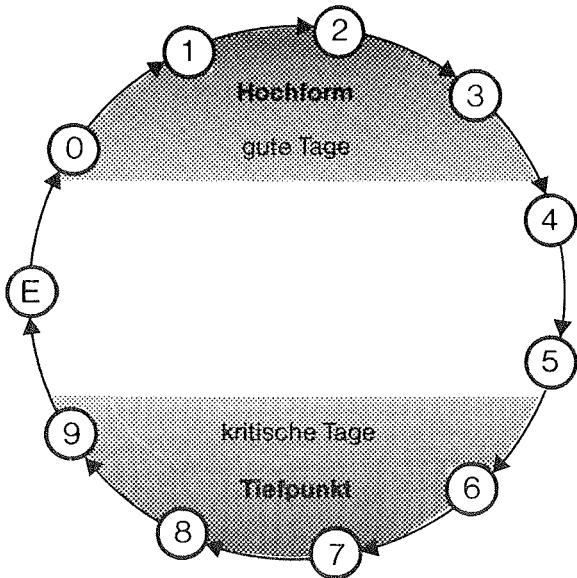
Programm-Start mit HALT – NEXT – 00. Das Display zeigt: 000000.

Als Beispiel wollen wir den Bio-Rhythmus eines Menschen berechnen, welcher am 2. 7. 1952 geboren ist. Die Berechnung soll uns zeigen, ob am 24. 12. 1982 bessere oder schlechtere Zyklus-Tage zu erwarten sind.

Wie bei den vorangegangenen Experimenten wird zunächst wieder das Geburtsjahr und Monat (195207) eingegeben. Nach Betätigung der Taste A warten bis das Display 00 zeigt, dann den Geburtstag (02 – 0 darf nicht vergessen werden) eingeben. Nun folgt in gleicher Weise das zweite interessierende Datum, zunächst wird Jahr und Monat (198212) eingegeben. Taste A betätigen – warten bis Display 00 bringt, dann Tag (24) eingeben.

Wenn wir die vorerwähnten Daten eingegeben haben, zeigt das Display jetzt **E**, als Ziffer für den körperlichen Rhythmus.

Bio-Kreis



Der Computer bringt nacheinander die Bio-Werte:

- Körperliche Zustands-Erwartung
- Seelische Zustands-Erwartung
- Geistige Zustands-Erwartung

Wenn wir den Bio-Kreis betrachten, finden wir das **E** auf der linken Seite, d. h. unsere Versuchsperson hat körperlich die kritischen Tage verlassen und nähert sich der Hochform-Zeit.

Wir betätigen die Taste 0 – das Display zeigt: **5**. Dies ist der Wert für die seelische Verfassung und wir sehen aus dem Bio-Kreis, daß sich der seelische Zustand in Richtung Tiefpunkt bewegt.

Nach abermaliger Tasten-Betätigung 0 zeigt das Display: **3**, den Wert für die geistige Verfassung. Der Bio-Kreis zeigt, daß die Versuchsperson für eine geistige Tätigkeit eine gute Kondition hat. Gemäß Lehre des Bio-Rhythmus könnte unsere Versuchsperson insgesamt mit einer sehr guten geistigen, körperlichen und seelischen Verfassung rechnen, wenn alle drei ermittelten Werte auf dem Bio-Kreis zwischen 0 und 4 liegen.

Nach nochmaliger Betätigung der 0-Taste ist der Computer für eine neue Biorhythmus-Berechnung bereit. Die Eingaben Reihenfolge ist grundsätzlich: Geburtsjahr und Geburtsmonat (evtl. 0-Eingabe nicht vergessen), Taste A, dann Geburtstag. In gleicher Weise wird das zweite zu berechnende Datum eingegeben.

Die zu berechnenden Daten dürfen nicht mehr als 273 Jahre auseinander liegen.

#### Das interessiert den Programmierer:

Register Nr.	Registerfunktionen
0 1 2 3 4 5	Rechenregister und Eingabe-Register für Tages-Datum
6 7	Speicher für die beiden ersten Stellen der Jahreszahl
8 9 A B C D	Faktorberechnung (Reg. 8 bis C) Speicher für Monatszahlen (Reg. C und D)
E F	Speicher für die beiden letzten Stellen der Jahreszahl

Bei den drei letzten Programm-Experimenten mußten Zeitberechnungen durchgeführt werden. Aus den eingegebenen Daten wird ein Faktor berechnet. Mit diesem Faktor werden die endgültigen Ergebnisse ermittelt.

Der Faktor berechnet sich nach folgender Formel, wenn das eingegebene Datum den Monat Januar oder Februar beinhaltet:

$$\text{Faktor} = 365 \times \text{Jahr} + \text{Tag} + 31 \times (\text{Monat} - 1) + \text{INT}((\text{Jahr} - 1) \div 4) - \text{INT}(3/4 \times \text{INT}((\text{Jahr} - 1) \div 1) + 1).$$

Die Bezeichnung INT (integer) besagt, daß bei dem Ergebnis der folgenden Berechnung die Stellen nach dem Komma nicht berücksichtigt werden sollen. Dies erleichtert unsere Programmierung.

Liegt das eingegebene Datum zwischen März und Dezember, berechnet sich der Faktor nach folgender Formel:

$$\text{Faktor} = 365 \times \text{Jahr} + \text{Tag} + 31 \times (\text{Monat} - 1) - \text{INT}(0,4 \times \text{Monat} + 2,3) + \text{INT}(\text{Jahr} \div 4) - \text{INT}(3/4 \times (\text{INT}(\text{Jahr} \div 100) + 1)).$$

Als Beispiel soll der Faktor für den 24. 12. 1982 errechnet werden. Wir müssen in die Formel für das Jahr **1982** für den Monat **12** und für den Tag **24** einsetzen:

$$\text{Faktor} = 365 \times 1982 + 24 + 31 \times (12 - 1) - \text{INT}(0,4 \times 12 + 2,3) + \text{INT}(1982 \div 4) - \text{INT}(3/4 \times (\text{INT}(1982 \div 100) + 1)).$$

$$\text{Faktor} = 723430 + 24 + 341 - 7 + 495 - 15.$$

$$\text{Faktor} = 724268$$

Beim Programm „Berechnung der Tage zwischen zwei Daten“ werden aus den beiden eingegebenen Daten zwei Faktoren berechnet. Die beiden Faktoren werden voneinander subtrahiert und das Ergebnis ist die Anzahl der Tage.

Das Prinzip ist also relativ einfach, es erfordert jedoch einen ziemlich großen Rechenaufwand.

Beim Programm „Bio-Rhythmus“ wird nach dem gleichen Prinzip die Anzahl der Tage zwischen Geburtsdatum und zu berechnendem Datum ermittelt. Dieses Ergebnis wird durch die Anzahl der Tage des entsprechenden Rhythmus geteilt (Rhythmus-Tage sind 23, 28 und 33). Der Rest dieser Division wird mit 10 multipliziert und dann durch die Anzahl der Rhythmus-Tage geteilt. Dieses Ergebnis bringt eine Ziffer zwischen 0 und 9 oder E, was auf dem Display angezeigt wird.

Bei der „Wochentag-Berechnung“ wird ebenfalls der Faktor aus dem eingegebenen Datum berechnet. Dieser Faktor muß dann nach folgender Formel weiterberechnet werden:

$$\text{Wochentag} = \text{Faktor} - (\text{INT}(\text{Faktor} \div 7) \times 7).$$

Setzen wir wieder den für das Datum 24. 12. 1982 errechneten Faktor in obige Formel ein, ergibt sich folgende Berechnung:

$$\text{Wochentag} = 724268 - (\text{INT}(724268 \div 7) \times 7).$$

$$\text{Wochentag} = 724268 - (\text{INT}(103466,8571) \times 7).$$

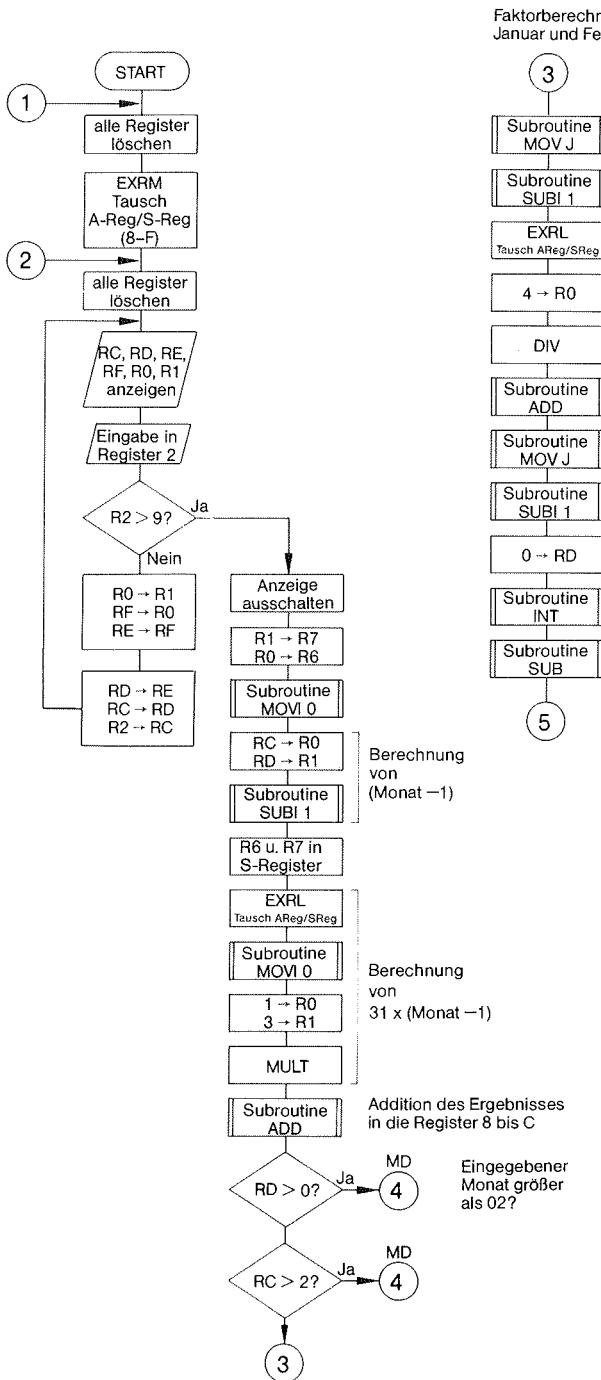
$$\text{Wochentag} = 724268 - (103466 \times 7).$$

$$\text{Wochentag} = 724268 - 724262 = 6$$

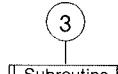
Wochentag = ein Freitag (6 – siehe Wochentag-Tabelle).

Obwohl auch diese Berechnung relativ einfach verständlich ist, ergibt sich ein aufwendiges Programm, weil viele Rechenschritte auszuführen sind. Zur Programm-Vereinfachung rechnet der Computer nur mit 5 Stellen. Beim Faktor: 724268 wird die erste Stelle (7) nicht mitgerechnet. Wir sehen jedoch aus den Darstellungen der Wochentags-Berechnung, daß sich die erste Stelle (7) nicht verändert und somit keinen Einfluß auf das Ergebnis hat.

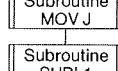
### Programm-Ablaufplan: Faktorberechnung



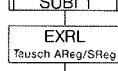
Faktorberechnung für Januar und Februar



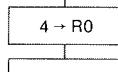
Berechnung von (Jahr - 1)



Berechnung von INT ((Jahr - 1) ÷ 4)

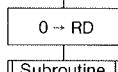
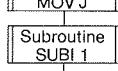
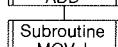
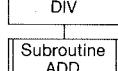


Addition des Ergebnisses in die Register 8 bis C

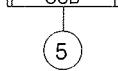
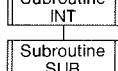


Gesamtberechnung von INT (¼ x (INT (Jahr - 1) ÷ 100) + 1)  
Subtraktion des Ergebnisses von den Registern 8 bis C

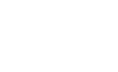
Berechnung von INT (Jahr - 1)



5



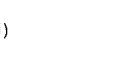
5



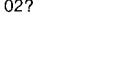
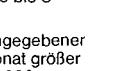
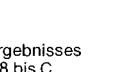
Berechnung von (Monat - 1)



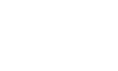
Berechnung von 31 x (Monat - 1)



Addition des Ergebnisses in die Register 8 bis C



R0 u. R1 anzeigen



Anzeige ausschalten



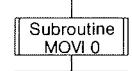
3

Eingegebener Monat größer als 02?

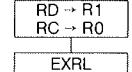
MD 4

MD 4

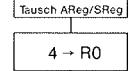
3



Berechnung von 4 x Monat



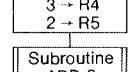
4 → R0



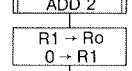
Addition von 23 zum Ergebnis der letzten Berechnung



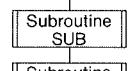
3 → R4  
2 → R5



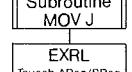
Subroutine ADD 2



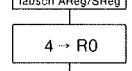
Bildung des INT-Wertes



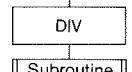
Subtraktion des Ergebnisses von INT (0,4 x Monat + 2,3) von R8 bis RC



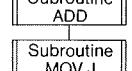
+ 2,3) von R8 bis RC



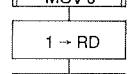
Berechnung von INT (Jahr ÷ 4)



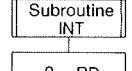
4 → R0



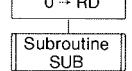
DIV



Addition des Ergebnisses zu den Registern 8-C



1 → RD



Subroutine INT

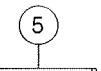


0 → RD

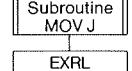


Subtraktion des Ergebnisses von den Registern 8-C

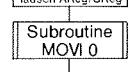
5



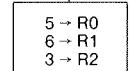
Berechnung von 365 x Jahre



EXRL Tausch AReg/SReg



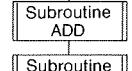
Subroutine MOVI 0



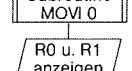
5 → R0  
6 → R1  
3 → R2



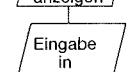
MULT



Addition zu den Registern 8-C



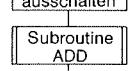
Subroutine MOVI 0



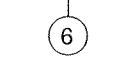
R0 u. R1 anzeigen



Eingabe in R0 u. R1



Anzeige ausschalten

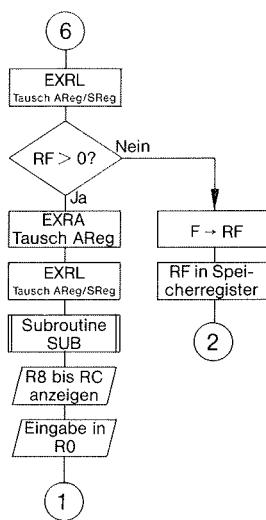


Addition der Eingabe zu den Registern 8-C

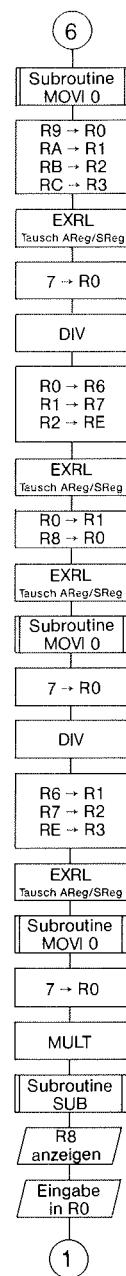
6

## Programm-Ablaufpläne:

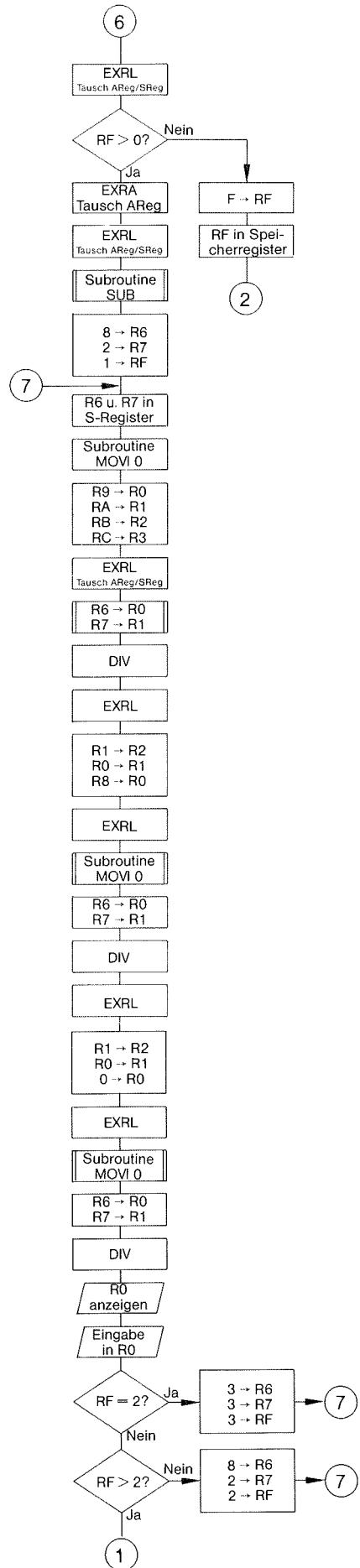
### Berechnung der Tage zwischen zwei Daten



### Berechnung eines Wochentages



### Berechnung des Bio-Rhythmus



Der größte Programm-Aufwand ist für die Berechnung des „Faktors“ notwendig (siehe auch Programm-Ablaufplan). Um den Programm-Aufwand möglichst gering zu halten, werden einige Unter-Programme verwendet, die wir bereits bei anderen Programmen kennengelernt haben. Diese Unter-Programme sind deshalb auch beim Programm-Ablaufplan nicht noch einmal aufgeführt. Der Ordnung halber sollen sie jedoch kurz erwähnt werden:

Das Unter-Programm (Subroutine) **SUB** führt eine 5-stellige Subtraktion durch. Die Register 0 bis 4 werden von den Registern 8 bis C abgezogen. In den Registern 8 bis C steht am Ende das Ergebnis. Alle Berechnungen werden dezimal ausgeführt. Die dezimale Subtraktion wurde beim Programm „Taschen-Rechner“ besprochen.

Das Unter-Programm **ADD** addiert die Register 0 bis 4 zu den Registern 8 bis C. Das Unter-Programm ADD führt eine 5-stellige dezimale Addition durch.

Das Unter-Programm **SUBI** addiert 1 zu einer Zahl, die in den Registern 0 bis 3 vorhanden ist.

Das Unter-Programm **ADD 2** addiert die Register 4 und 5 dezimal zu den Registern 1 und 2.

Das Unter-Programm **MOVI 0** löscht die Register 0 bis 5.

Das Unter-Programm **MOVJ** (move-Jahre) bringt die in den Registern 6, 7, E und F zwischengespeicherten Jahreszahlen in die Register 0 bis 3.

Der Programm-Teil **INT** berechnet den Formel-Teil INT ( $3/4 \times (\text{INT}(\text{Jahre} \div 100) + 1)$ ).

Die einzelnen Formel-Abschnitte des Programmteils INT werden nacheinander berechnet und in die Register 8 bis C addiert, bzw. subtrahiert. Als Ergebnis dieser Berechnungen steht dann in den Registern 8 bis C der Faktor, welcher durch das folgende Programm weiter bearbeitet werden kann.

### Ein wichtiger Hinweis für Programmierer!

Die Unter-Programme **ADD** (5-stellige dezimale Addition) und **SUB** (5-stellige dezimale Subtraktion) können für vielerlei selbst zu entwickelnder Programme verwendet werden.

Das Unter-Programm ADD haben wir beim Programm „Berechnung der Tage zwischen zwei Daten“ von Adresse BC bis Adresse DE eingegeben. Das Unter-Programm SUB beginnt bei der Adresse DF und endet bei FF.

Beide Unter-Programme stehen am Ende des Programm-Speichers und können (wenn der Computer nicht vom Netzstrom getrennt wird), jederzeit aufgerufen werden. Wir erinnern uns, daß wir mit dem Befehl CALL von einem Programm in ein Unter-Programm springen können. Mit dem Befehl CALL BC erreichen wir das Unter-Programm ADD. Mit dem Befehl CALL DF das Unter-Programm SUB.

### **3. Teil:**

## **Computer-Experimente mit zusätzlichen Elektronik-Bauelementen**

Die Möglichkeiten des Computers werden durch zusätzlich angeschlossene Bauelemente wesentlich erweitert. Solche Zusatz-Schaltungen nennt man „Peripherie-Elektronik“.

### **Bei Schaltungs-Experimenten ist unbedingt zu beachten:**

Beim Experimentieren dürfen **in keinem Fall Spannungen über 9V Gleichstrom** eingesetzt werden. Deshalb grundsätzlich mit der vorgesehenen 9 Volt Batterie oder mit dem BUSCH Netzgerät 2059 (welches ebenfalls 9 Volt Gleichstrom abgibt) experimentieren. Spannungen über 9 Volt und ganz besonders Wechselstrom-Spannungen können zur Zerstörung einzelner Computer-Elemente führen.

**Die Computer-Ausgänge dürfen niemals direkt mit dem Plus- oder Minus-Pol einer 9 Volt Spannungsquelle (Batterie oder Netzgerät) verbunden werden.**

An den Computer-Ausgängen stehen lediglich kleine Steuerspannungen (maximal 15 mA) zu Verfügung. Deshalb können an den Computer-Ausgängen nur Elemente mit geringen Anschlußwerten direkt betrieben werden, wie z. B. Piezo-Summer, Leuchtdioden, Transistoren, IC's usw. Mit einem Transistor und 1 bis 2 Widerständen können jedoch sehr einfach kleine Verstärker-Schaltungen aufgebaut werden, wodurch dann auch andere Schaltungsteile durch die Computer-Ausgänge gesteuert werden können. Auf den folgenden Seiten werden verschiedene Anschlußmöglichkeiten demonstriert.

Da auch an den Computer-Eingängen nur geringe Spannungen für die Eingangs-Signale benötigt werden, ist die Verwendung eines Vorwiderstandes in jedem Fall empfehlenswert, weil sich hierdurch die Lebensdauer der angeschlossenen Batterie erheblich verlängert. Auf Seite 54 des Anleitungsbuches 1. Teil wurde die Anordnung solcher Vorwiderstände (4,7 kΩ) in Verbindung mit den roten Tasten dargestellt.

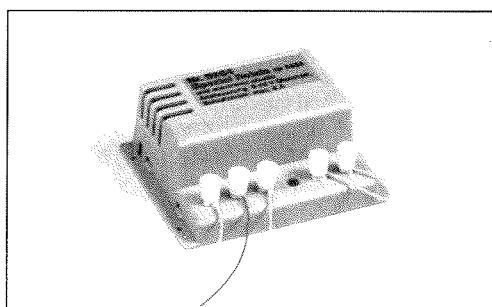
# Anschluß von Relais

Aus verschiedenen Experimenten (Anleitungsbuch 1. Teil) wissen wir, daß der Computer Schaltfunktionen übernehmen kann. Hierfür bringen wir mit dem DOT-Befehl und durch entsprechende Programmierung „eine Spannung“ an die Computer-Ausgänge. Mit dieser geringen Steuerspannung können wir jedoch z. B. nur den Piezo-Summer oder eine Leuchtdiode betreiben.

Größere Stromverbraucher erhalten ihre Spannungsversorgung über ein Schaltrelais. Das Schaltrelais wird vom Computer-Ausgang angesteuert. Da auch die Relais-Funktion mehr Strom bzw. mehr Spannung benötigt als am Computer-Ausgang zur Verfügung steht, muß die Ausgangs-Spannung durch eine Transistorschaltung verstärkt werden.

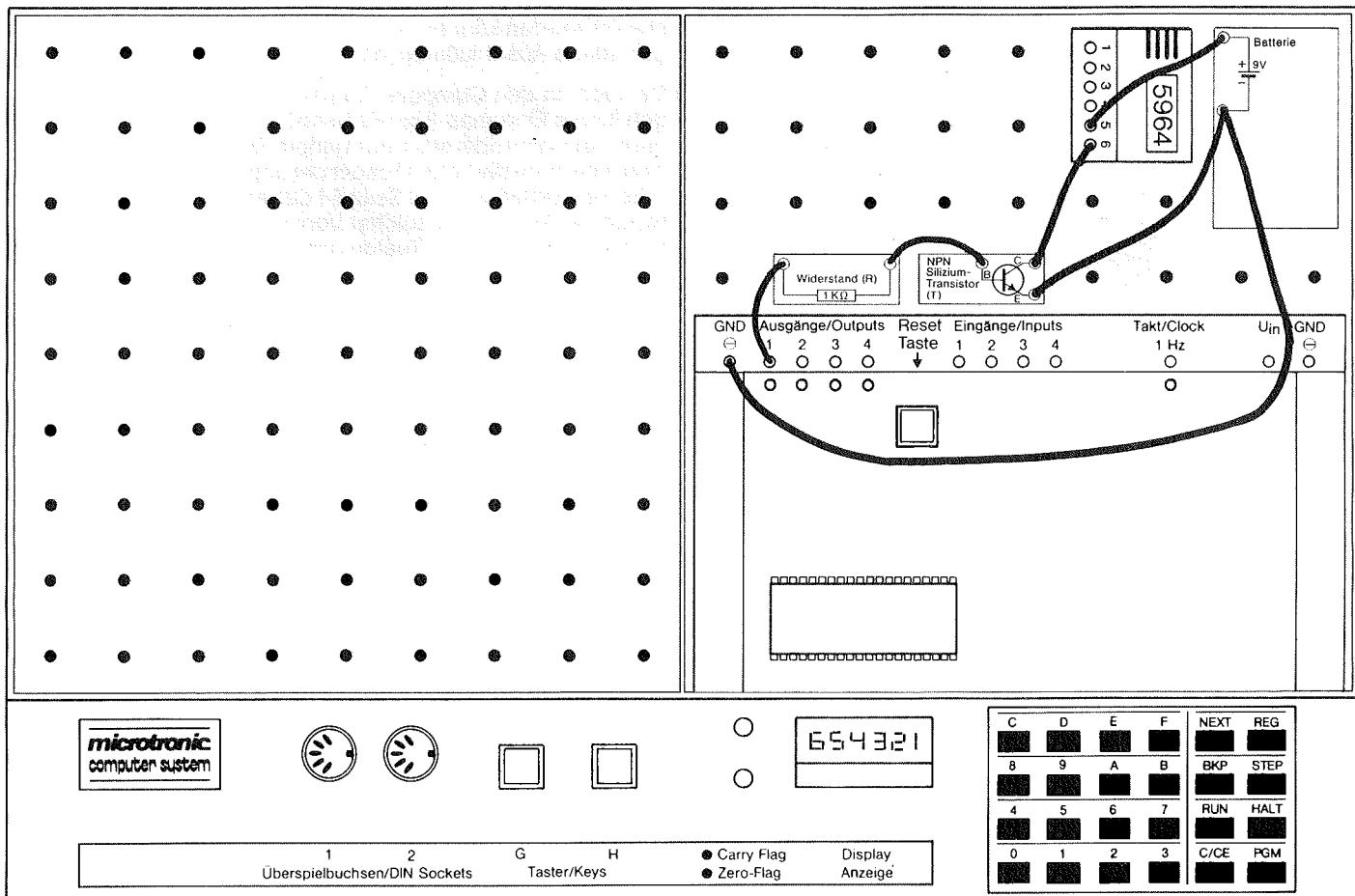
Zwei Relais-Möglichkeiten stehen zur Verfügung:

## BUSCH-Schwachstrom-Schaltrelais Nr. 5964



Dieses Relais wird u. a. auch im BUSCH-Modellbahn-Zubehör-Programm angeboten. Es ist für vielseitige Schalt- und Steuerfunktionen verwendbar.

## Anschluß des Relais 5964 an einen Computer-Ausgang



Da jeder Computer-Ausgang durch entsprechende Programmierung einzeln angesteuert werden kann, ist der Anschluß von 4 Relais möglich. Zu jedem Relais ist jeweils ein Widerstand 1 KΩ und ein Transistor erforderlich.

### BUSCH-Netzstrom-Schaltgerät 2087



Neben vielen anderen Einsatzmöglichkeiten kann der Computer in Verbindung mit BUSCH-Elektronik-Studios und Schaltgerät 2087 z. B. während eines Ferien-Aufenthaltes die Anwesenheit der Bewohner eines Hauses vortäuschen. Ein Fotowiderstand schaltet bei Einbruch der Dunkelheit eine Beleuchtung an, die sich im Morgengrauen wieder abschaltet. Eine geräuschempfindliche Schaltung sorgt für eine vorübergehende Zusatzbeleuchtung, sobald ein bestimmter Geräuschpegel überschritten wird. Der als Schaltuhr programmierte Computer kann zusätzlich zwei weitere Ein- und Ausschalt-Vorgänge übernehmen, wobei getrennte Zeiten einstellbar sind. Eine abgesicherte Tür löst einen Alarmton aus, sobald die Tür geöffnet wird.

Die folgenden Programm-Beispiele sollen lediglich Anregungen geben. Es würde den Rahmen dieses Anleitungsbuches übersteigen, wenn alle Anwendungsmöglichkeiten aufgezählt und erläutert werden sollten. Den Besitzern zusätzlicher BUSCH-Electronic-Studios eröffnen sich unzählige Möglichkeiten.

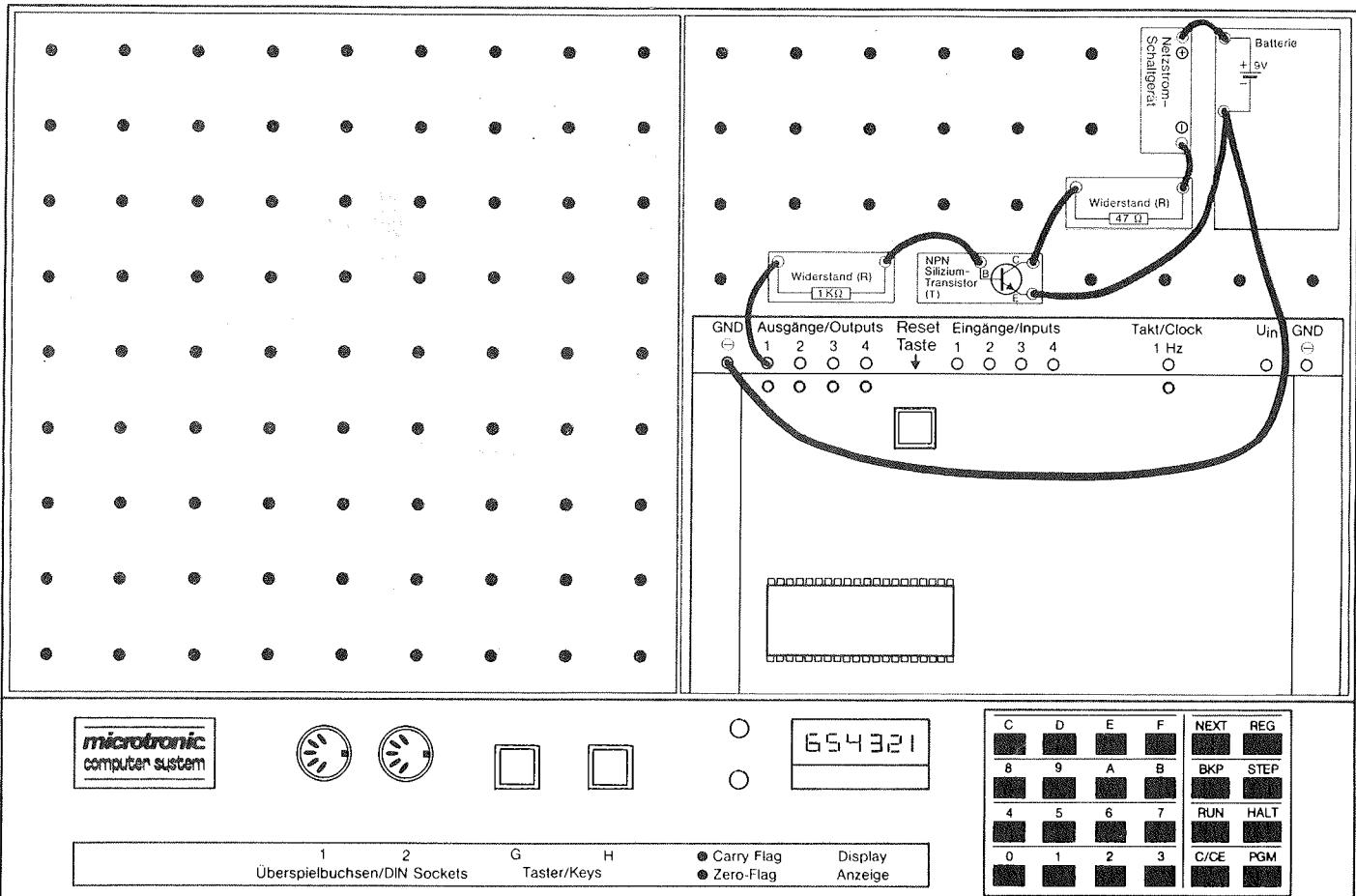
Auch dieses Schaltgerät kann mit Schwachstrom (6-9 Volt) angesteuert werden. Es löst über ein Relais Schaltvorgänge im Lichtstromnetz (220 V, max. 1000 Watt) aus. Mit zwei Widerständen und einem Transistor (siehe Abbildung) wird das Schaltgerät an einem Computer-Ausgang angeschlossen.

Das Schaltgerät 2087 entspricht den VDE Vorschriften. Es ist absolut gefahrlos einzusetzen, weil der 220 V Netzstromteil von der Schwachstromseite völlig getrennt ist.

Das Schaltgerät 2087 ist ein „Ein-/Ausschalter“. Sobald eine geringe Steuerspannung vorhanden ist, schließt das Relais einen 220 V Schaltkreis. Dieser Stromkreis wird wieder abgeschaltet, wenn die Steuerspannung unterbrochen wird. Das Relais ist für 20 Millionen Schaltvorgänge ausgelegt und kann im Dauereinsatz betrieben werden. Es dient zum Ein- und Ausschalten von Radios, Tonbandgeräten, Warnanlagen, Heizlüftern und allen elektrischen Geräten mit einem Anschlußwert bis zu 1000 Watt. Werden Geräte mit höherem Anschlußwert geschaltet, muß mit einer erheblichen Verringerung der Schaltleistung gerechnet werden, es ist deshalb z. B. beim Anschluß von Heizlüftern usw. unbedingt darauf zu achten, daß die 1000 Watt Grenze nicht überschritten wird.

Bis zu 4 Netzstrom-Schaltgeräte 2087 können an den Computer-Ausgängen angeschlossen werden. Für jedes Schaltgerät sind zwei Widerstände und ein Transistor erforderlich.

### Anschluß des Netzstrom-Schaltgeräts 2087 an einen Computer-Ausgang



## Schaltuhr mit zwei Ein- und Ausschalt-Zeiten

Mit dem folgenden Programm können zwei getrennte Ein- und Ausschalt-Zeiten erreicht werden. Hierdurch sind zwei unterschiedliche Funktionen zu unterschiedlichen Zeiten möglich.

Nach HALT – NEXT – 00 das Programm eingeben:

### Programm: Schaltuhr für zwei Ein- und Ausschalt-Zeiten

Sprungziel	Adr. Nr.	Eingabe-Befehl	Mnemonic	Sprung nach	Erklärungen
	00	<b>F08</b>	CLEAR		Arbeitsregister löschen
	01	<b>F0D</b>	EXRL		Arbeitsregister/Speicherregister tauschen
	02	<b>F08</b>	CLEAR		Arbeitsregister löschen
	03	<b>FE8</b>	DOT 8		Ausgänge abschalten
„START“	04	<b>108</b>	MOVI 0,8		
	05	<b>F18</b>	DISP 1,8		8 anzeigen
	06	<b>FF8</b>	KIN 8		Eingabe in R8
	07	<b>9A8</b>	CMPI A,8		
	08	<b>E3A</b>	BRZ 3A „S UHR“		
	09	<b>9B8</b>	CMPI B,8		
	0A	<b>E32</b>	BRZ 32 „UHR“		
	0B	<b>9C8</b>	CMPI C,8		
	0C	<b>E13</b>	BRZ 13 „EIN 1“		
	0D	<b>9D8</b>	CMPI D,8		
	0E	<b>E17</b>	BRZ 17 „AUS 1“		
	0F	<b>9E8</b>	CMPI E,8		
	10	<b>E1B</b>	BRZ 1B „EIN 2“		
	11	<b>D1D</b>	BRC 1D „AUS 2“		
	12	<b>C04</b>	GOTO 04 „START“		
„EIN 1“	13	<b>F0D</b>	EXRL		Speicherregister/Arbeitsregister tauschen
	14	<b>B1F</b>	CALL 1F „S EIN“		
	15	<b>F0D</b>	EXRL		Arbeitsregister/Speicherregister tauschen
	16	<b>C04</b>	GOTO 04 „START“		
„AUS 1“	17	<b>F0D</b>	EXRL		Speicherregister/Arbeitsregister tauschen
	18	<b>B29</b>	CALL 29 „S AUS“		
	19	<b>F0D</b>	EXRL		Arbeitsregister/Speicherregister tauschen
	1A	<b>C04</b>	GOTO 04 „START“		
„EIN 2“	1B	<b>B1F</b>	CALL 1F „S EIN“		
	1C	<b>C04</b>	GOTO 04 „START“		
„AUS 2“	1D	<b>B29</b>	CALL 29 „S AUS“		
	1E	<b>C04</b>	GOTO 04 „START“		
„S EIN“	1F	<b>F40</b>	DISP 4,0		Unterprogramm „S EIN“
	20	<b>FF8</b>	KIN 8		Eingabe der Einschaltzeit
	21	<b>998</b>	CMPI 9,8		
	22	<b>D28</b>	BRC 28		
	23	<b>023</b>	MOV 2,3		
	24	<b>012</b>	MOV 1,2		
	25	<b>001</b>	MOV 0,1		
	26	<b>080</b>	MOV 8,0		
	27	<b>C1F</b>	GOTO 1F		
	28	<b>F07</b>	RET		
„S AUS“	29	<b>F44</b>	DISP 4,4		Unterprogramm „S AUS“
	2A	<b>FF8</b>	KIN 8		Eingabe der Ausschaltzeit
	2B	<b>998</b>	SMPI 9,8		
	2C	<b>D28</b>	BRC 28		
	2D	<b>067</b>	MOV 6,7		
	2E	<b>056</b>	MOV 5,6		
	2F	<b>045</b>	MOV 4,5		
	30	<b>084</b>	MOV 8,4		
	31	<b>C29</b>	GOTO 29		

„UHR“	32	<b>108</b>	MOVI 0,8		Uhrenbetrieb ohne Schaltfunktion
	33	<b>FE8</b>	DOT 8		Ausgänge ausgeschaltet
	34	<b>F6A</b>	DISP 6,A		
	35	<b>F06</b>	TIME		
	36	<b>FD8</b>	DIN 8		
	37	<b>FA8</b>	SHL 8		
	38	<b>E34</b>	BRZ 34		
	39	<b>C04</b>	GOTO 04 „START“		
„S UHR“	3A	<b>F4C</b>	DISP 4,C		Schaltuhrenbetrieb
	3B	<b>F06</b>	TIME		
	3C	<b>B65</b>	CALL 65 „V EIN“		
	3D	<b>D40</b>	BRC 40		
	3E	<b>12A</b>	MOVI 2,A		
	3F	<b>AA9</b>	OR A,9		
	40	<b>B74</b>	CALL 74 „V AUS“		
	41	<b>D43</b>	BRC 43		
	42	<b>3D9</b>	ANDI D,9		
	43	<b>F0D</b>	EXRL		
	44	<b>B65</b>	CALL 65 „V EIN“		
	45	<b>D48</b>	BRC 48		
	46	<b>11A</b>	MOVI 1,A		
	47	<b>AA9</b>	OR A,9		
	48	<b>B74</b>	CALL 74 „V AUS“		
	49	<b>D4B</b>	BRC 4B		
	4A	<b>3E9</b>	ANDI E,9		
	4B	<b>F0D</b>	EXRL		
	4C	<b>FD8</b>	DIN 8		Eingabe von Taster G u. H in R8
	4D	<b>338</b>	ANDI 3,8		Taktimpuls aus R8 löschen
	4E	<b>E60</b>	BRZ 60 „OR“		
	4F	<b>938</b>	CMPI 3,8		
	50	<b>E04</b>	BRZ 04 „START“		
	51	<b>09A</b>	MOV 9,A		
	52	<b>918</b>	CMPI 1,8		
	53	<b>E5B</b>	BRZ 5B „TA-G“		
„TA-H“	54	<b>F9A</b>	SHR A		Auswertung Taster H
	55	<b>F9A</b>	SHR A		
	56	<b>D59</b>	BRC 59		
	57	<b>D29</b>	ADDI 2,9		
	58	<b>C60</b>	GOTO 60 „OR“		
	59	<b>3D9</b>	ANDI D,9		
	5A	<b>C60</b>	GOTO 60 „OR“		
„TA-G“	5B	<b>F9A</b>	SHR A		Auswertung Taster G
	5C	<b>D5F</b>	BRC 5F		
	5D	<b>519</b>	ADDI 1,9		
	5E	<b>C60</b>	GOTO 60 „OR“		
	5F	<b>3E9</b>	ANDI E,9		
„OR“	60	<b>339</b>	ANDI 3,9		Sind Ausgänge 1 und 2 „low“ (0), setzen Zero-Flag, somit wird Ausgang 3 ebenfalls „low“
	61	<b>E63</b>	BRZ 63		
	62	<b>549</b>	ADDI 4,9		
	63	<b>FE9</b>	DOT 9		
	64	<b>C3A</b>	GOTO 3A „S UHR“		
„V EIN“	65	<b>83F</b>	CMP 3,F		10-Stunden-Vergleich
	66	<b>E68</b>	BRZ 68		
	67	<b>C70</b>	GOTO 70		

	68	<b>82E</b>	CMP 2,E		Stunden-Vergleich
	69	<b>E6B</b>	BRZ 6B		
	6A	<b>C70</b>	GOTO 70		
	6B	<b>81D</b>	CMP 1,D		10-Minuten-Vergleich
	6C	<b>E6E</b>	BRZ 6E		
	6D	<b>C70</b>	GOTO 70		
	6E	<b>80C</b>	CMP 0,C		Minuten-Vergleich
	6F	<b>E72</b>	BRZ 72		
	70	<b>F09</b>	STC		Carry-Flag setzen (1)
	71	<b>F07</b>	RET		
	72	<b>F0A</b>	RSC		Carry-Flag rücksetzen (0)
	73	<b>F07</b>	RET		
„V AUS“	74	<b>87F</b>	CMP 7,F		10-Stunden-Vergleich
	75	<b>E77</b>	BRZ 77		
	76	<b>C70</b>	GOTO 70		
	77	<b>86E</b>	CMP 6,E		Stunden-Vergleich
	78	<b>E7A</b>	BRZ 7A		
	79	<b>C70</b>	GOTO 70		
	7A	<b>85D</b>	CMP 5,D		10-Minuten-Vergleich
	7B	<b>E7D</b>	BRZ 7D		
	7C	<b>C70</b>	GOTO 70		
	7D	<b>84C</b>	CMP 4,C		Minuten-Vergleich
	7E	<b>E72</b>	BRZ 72		
	7F	<b>C70</b>	GOTO 70		

Nach der Programm-Eingabe müssen an den EINGÄNGEN 1 und 2 die beiden roten Taster G und H, außerdem die Verbindungsleitung TAKT/CLOCK zum EINGANG 4 angeschlossen werden. Der Piezo-Summer wird mit dem AUSGANG 3 und GND verbunden. Die Anschlußleitungen des Piezo-Summers dürfen nicht vertauscht werden. (Siehe Abbildung).

Nach den Tasten HALT – PGM – 3 die momentane Uhrzeit eingeben. Dann Programm-Start mit HALT – NEXT – 00 – RUN. Das Display zeigt 0. Nun ergeben sich folgende Möglichkeiten:

**Taste B:** Auf dem Display wird die Uhr angezeigt. Die Schaltuhr arbeitet noch ohne Schaltfunktionen. Wird eine der beiden roten Tasten G oder H betätigt, zeigt das Display wieder 0, der Computer wartet auf die nächste Eingabe.

**Taste C:** Die erste Einschaltzeit kann eingegeben werden (Stunden und Minuten). Wurde versehentlich eine falsche Zeit eingegeben, richtige Eingabe wiederholen, bis die gewünschten Stunden und Minuten auf dem Display angezeigt werden. Eingabe abschließen durch nochmalige Betätigung der Taste C. Das Display zeigt wieder 0 – Computer wartet auf neue Eingabe.

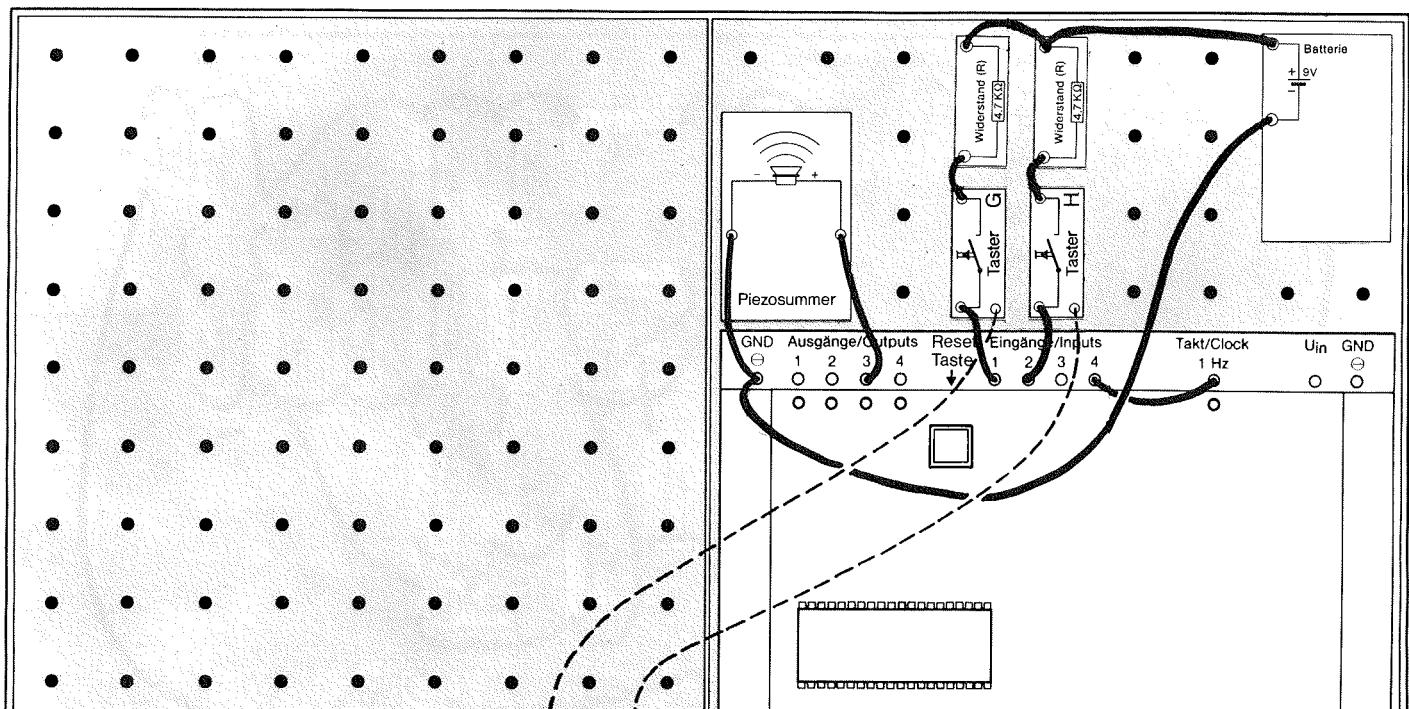
**Taste D:** Eingabe der ersten Ausschaltzeit (Stunden und Minuten). Eingabe durch nochmalige Betätigung der Taste D ab schließen. Display zeigt wieder 0.

**Taste E:** Eingabe der zweiten Einschaltzeit. Durch nochmaligen Tastendruck E Eingabe beenden.

**Taste F:** Eingabe der zweiten Ausschaltzeit. Eingabe durch nochmaligen Tastendruck F beenden.

**Taste A:** Die Schaltuhr-Funktionen werden in Betrieb genommen. Das Display zeigt die momentane Uhrzeit in Stunden und

#### Anschluß der Taster u. Piezosummer für Schaltuhr



**microtronic**  
computer system



654321  
● Carry Flag  
● Zero-Flag



Display  
Anzeige



Überspielbuchsen/DIN Sockets

G H  
Taster/Keys

Minuten (ohne Sekunden). Der Piezo-Summer meldet sich, sobald die erste Einschaltzeit erreicht wird. Mit der roten Taste G kann der Summer abgestellt werden. Die rote Taste ist ca. 1 Sekunde gedrückt zu halten. Während der ersten Einschalt-Minute meldet sich der Summer allerdings nach einer Sekunde erneut, d. h. daß während der Einschalt-Minute die rote Taste ständig gedrückt sein muß, wenn der Summer nicht ertönen soll. Sobald auf dem Display die nächste volle Minute angezeigt wird, kann mit der roten Taste G der Summer wahlweise endgültig abgeschaltet oder auch eingeschaltet werden. Diese Einschaltmöglichkeit, unabhängig von der eingestellten Zeit, ist für die folgenden Versuche interessant.

Wenn die erste Einschaltzeit durch Tastendruck G abgestellt wurde, meldet sich der Computer erneut, sobald die zweite Einschaltzeit erreicht wird. Der Summer kann jetzt mit der roten Taste H abgeschaltet werden, wobei ebenfalls während der eingestellten Schaltminute die Taste H ständig gedrückt sein muß, um das Summen abzustellen. Nach Ablauf der Schaltminute dient die Taste H ebenfalls dazu, den Summer endgültig ab- oder anzustellen.

Bei der ersten Einschaltzeit leuchten die LED's am Ausgang 1 und 3. Bei der zweiten Einschaltzeit leuchten die LED's an den Ausgängen 2 und 3. Mit der roten Taste G können somit die Ausgänge 1 und 3 und mit der Taste H die Ausgänge 2 und 3 ein- bzw. ausgeschaltet werden.

### Praktische Anwendung: Weck-Uhr mit 2 Weck-Zeiten

Unsere Schaltuhr ist ein idealer Wecker für alle, die am frühen Morgen nicht sofort beim ersten Ton aus den Federn springen.

Beispiel: Mit Taste C die erste Weckzeit 06.30 Uhr einstellen.

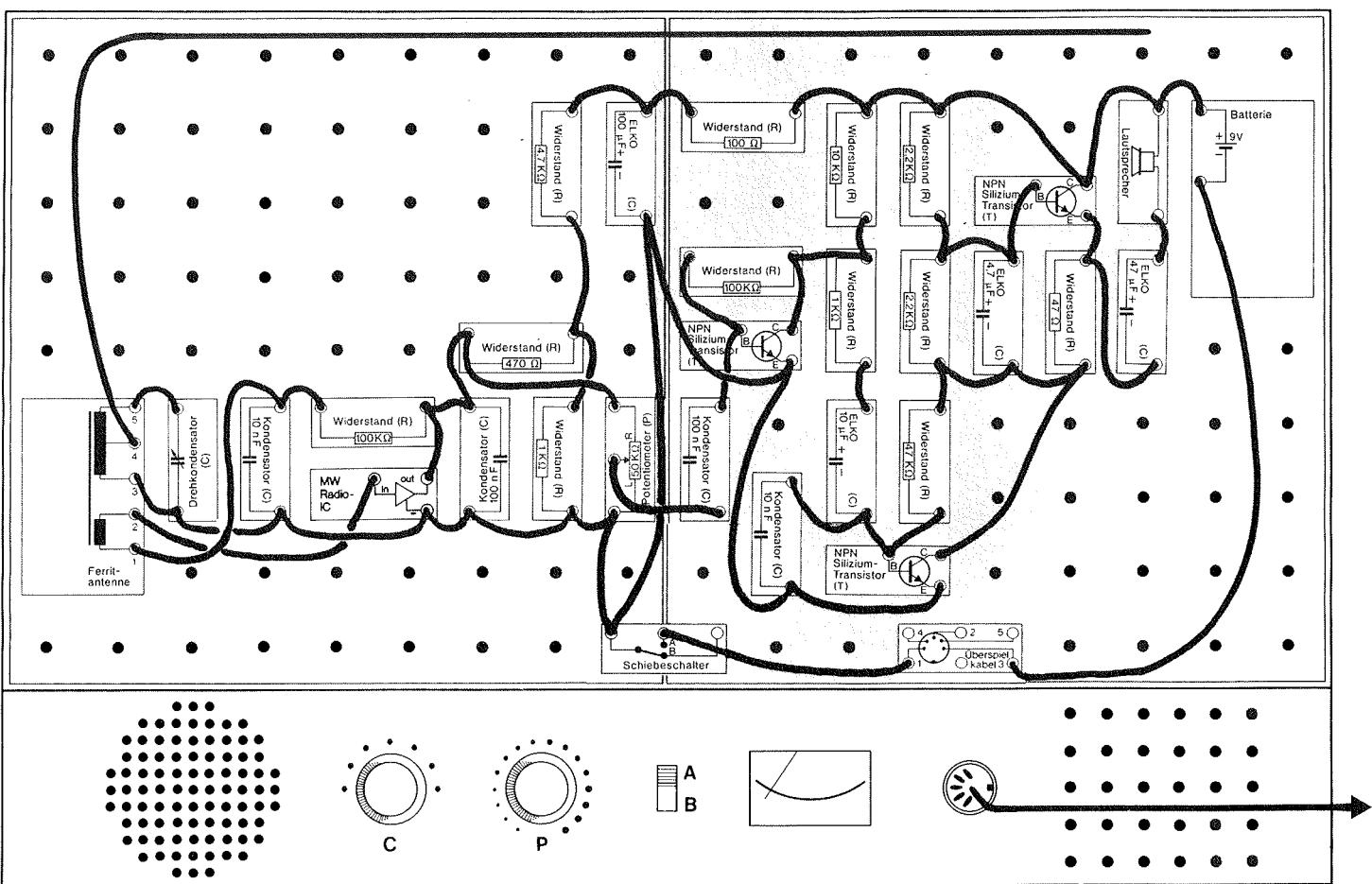
Mit Taste D die erste Ausschaltzeit 06.31 Uhr eingeben. Mit Taste E die zweite Weckzeit 06.45 Uhr eingeben. Mit Taste F das Ende der zweiten Weckzeit 07.00 Uhr eingeben. Weckuhr mit Taste A in Betrieb nehmen.

Der Wecker meldet sich morgens 6.30 Uhr. Er wird eine Minute lang summen. Wer weiterschlafen will, muß in der Zeit 6.30 Uhr bis 6.31 die Taste G gedrückt halten. Wenn das Summen nicht stört, wird sich die Weckuhr um 6.31 Uhr selbsttätig abschalten. Um 6.45 Uhr beginnt das zweite Wecken. Wenn die Taste H nicht betätigt wird, wird der Wecker 15 Minuten lang summen und schaltet sich dann selbsttätig ab. Werden die Weckzeiten nicht geändert, meldet sich der Computer am nächsten Morgen zur eingestellten Zeit erneut.

### Ändern der Schaltzeiten

Zum Ändern der Schaltzeiten müssen beide Tasten G und H gleichzeitig gedrückt werden. Das Display zeigt wieder 0 und wir können jetzt mit den Tasten C, D, E und F die eingestellten Zeiten anzeigen lassen, oder durch Neueingabe verändern. Nach Abschluß der Anzeige oder der Änderung wird jeweils noch einmal die entsprechende Buchstaben-Taste betätigt. Mit Taste A wird die Schaltuhr wieder in Betrieb genommen.

### MW-Radio zum Anschluß an Schaltuhr



## Computer-gesteuerte Schaltuhr mit vielen Variationsmöglichkeiten

Für die folgenden Experimente wird das im vorangegangenen Kapitel eingegebene Programm „Schaltuhr“ weiter verwendet. Durch Anschluß zusätzlicher Bauelemente ergeben sich beispielsweise folgende Variationen:

### Radio-Wecker

(BUSCH-Electronic-Studio 2070 erforderlich).

Mit dem BUSCH-Studio-Center 2070 wird ein Rundfunkempfänger aufgebaut. (Siehe Abbildung). Bei den Radio-Schaltungen führt normalerweise das Anschlußkabel vom Minus-Pol der Batterie zum Schiebeschalter des Electronic-Studios. Nachdem das Radio vom Computer eingeschaltet werden soll, wird der Baustein „Überspielkabel“ dazwischen geschaltet wie dies aus der Abbildung ersichtlich ist. Wir können jetzt das Electronic-Center 2070 mit aufgebautem Radio an den Computer anschließen, indem die beiden Geräte durch ein genormtes Überspielkabel (in jedem Radio-Geschäft erhältlich) miteinander verbunden werden. Dieses Überspielkabel führt von der Überspielbuchse des Studio-Centers zur Überspielbuchse Nr. 1 des Computers.

Im Computer-Gehäuse wird mit einem Transistor und dem  $1\text{ k}\Omega$  Widerstand eine einfache Verstärker-Schaltung aufgebaut, die mit dem Baustein „Überspielkabel 1“ verbunden ist. Da wir mit der geringen Spannung des Computer-Ausgangs unser Radiogerät nicht betreiben können, ist der Aufbau des kleinen Transistor-Verstärkers notwendig. Die Transistor-Verstärkerschaltung wird am AUSGANG 2 des Computers angeschlossen. Der Piezo-Summer wird mit dem AUSGANG 1 des Computers verbunden. Die Anordnung der Taster bleibt wie beim vorangegangenen Versuch bestehen. Die Abbildung zeigt alle erforderlichen Schaltungsaufbauten und Änderungen.

Die Eingabe der Ein- und Ausschaltzeiten sowie die Inbetriebnahme entspricht auch weiterhin der Anleitung im Kapitel „Schaltuhr“.

Es ist logisch, daß wir das aufgebaute Rundfunkgerät auf einwandfreie Funktion überprüfen, bevor wir den Baustein „Überspielkabel“ dazwischenschalten.

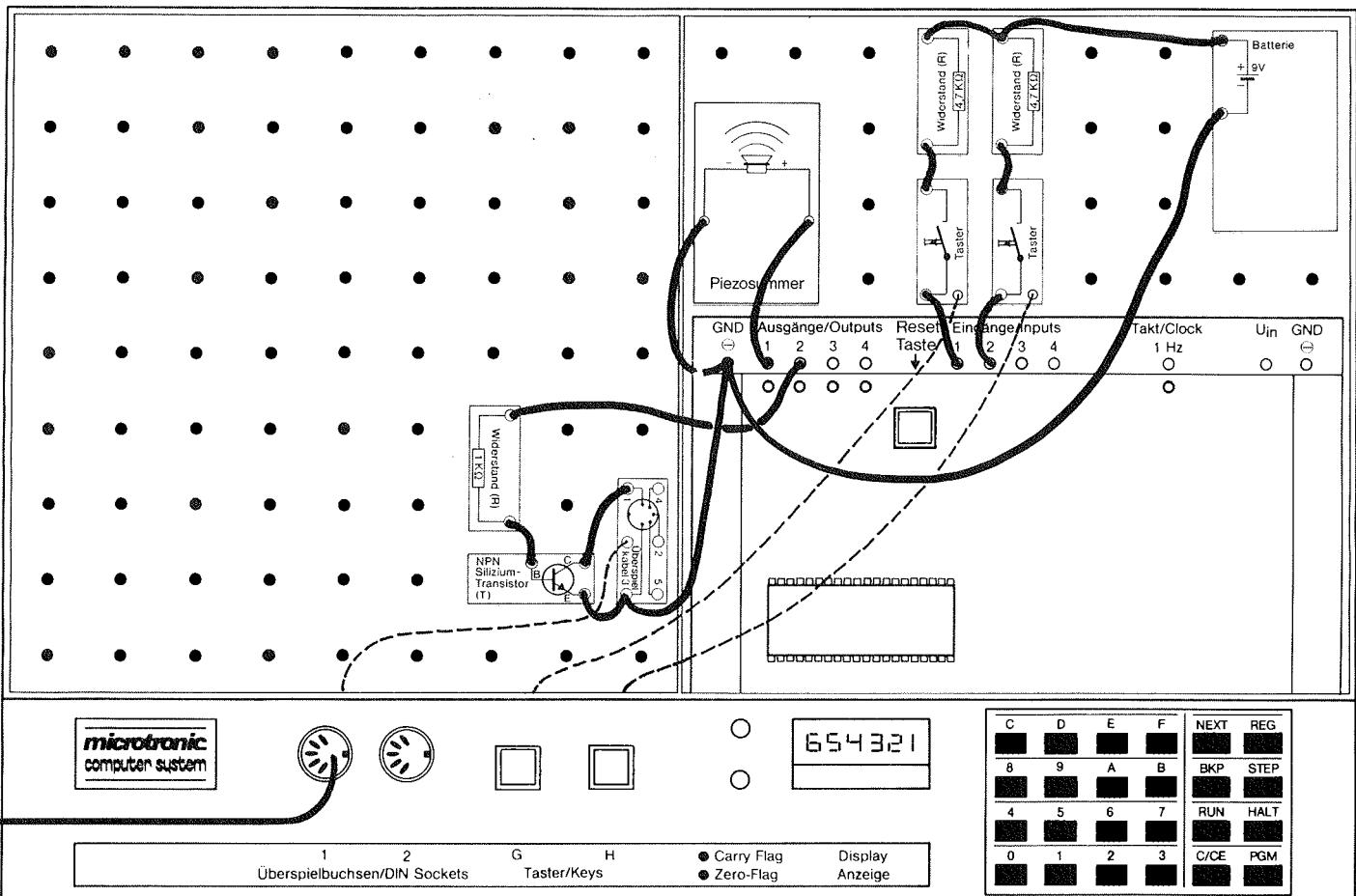
Die Eingabe der ersten Weckzeit wird jetzt den Piezo-Summer auslösen, während die zweite Weckzeit den Rundfunkempfänger einschaltet.

Wir können die beiden zusammengeschalteten Geräte (Computer mit Rundfunkempfänger) nun folgendermaßen betreiben:

Als erste Weckzeit wird wieder 06.30 Uhr eingegeben. Als erste Abschaltzeit 06.31 Uhr. Als zweite Einschaltzeit geben wir wieder 06.45 Uhr ein und als zweite Ausschaltzeit 07.15 Uhr.

Der Computer meldet sich morgens um 6.30 Uhr mit einem eine Minute andauernden Summton. Nach einer Pause von 14 Minuten wird um 6.45 Uhr das Radiogerät eingeschaltet. Wird der Rundfunkempfänger nicht durch Betätigung einer roten Taste abgeschaltet, erfolgt automatische Abschaltung um 7.15 Uhr.

### Schaltuhr für MW-Radio-Anschluß



Wenn wir abends gerne bei leiser Musik einschlafen, jedoch einige Stunden später wieder wach werden, weil das Radio immer noch läuft, können wir unser Wecker-Radio durch geänderte Zeiteingabe zum Schlummer-Radio ändern. Hierfür bleibt die gesamte Anordnung des vorstehend beschriebenen Wecker-Radios bestehen. Wir nehmen lediglich eine Zeitänderung vor.

Beispiel: Wir gehen um 22.00 Uhr ins Bett und das Radiogerät soll sich um 23.00 Uhr abschalten. Es wird lediglich die zweite Ausschaltzeit (Taste F) durch die Eingabe 23.00 Uhr geändert. Hierdurch schaltet unser Rundfunkempfänger nicht wie beim vorher beschriebenen Wecker-Radio um 7.15 Uhr automatisch ab, sondern erst um 23.00 Uhr.

Wenn wir um 22.00 Uhr durch Betätigung der roten Taste H einschalten, erfolgt die automatische Abschaltung um 23.00 Uhr. Da wir jedoch die übrigen Einschaltzeiten nicht geändert haben, wird unser „Schlummer-Radio-Wecker“ um 6.30 Uhr mit einem eine Minute dauernden Summton vorwecken und um 6.45 Uhr schaltet sich das Rundfunkgerät ein. Wenn wir aufgestanden sind genügt ein Druck auf die rote Taste H, um den Empfänger abzuschalten. Er wird am Abend ebenfalls mit Taste H wieder eingeschaltet und schaltet sich um 23.00 Uhr (oder zu jeder anderen eingegebenen Zeit) automatisch ab.

Wird das Radiogerät tagsüber nicht benötigt, können wir es zusätzlich am Schiebeschalter des Studio-Centers abschalten, weil unsere Transistor-Verstärker-Schaltung eine sehr minimale Betriebsspannung aus der Batterie entnimmt, die wir durch das Abschalten sparen können.

Übrigens, den gleichen Effekt können wir auch mit einem Netzstrom-Rundfunkempfänger oder einer Stereo-Anlage erreichen. Anstelle des selbstgebauten Radiogeräts und der Transistor-Schaltung verwenden wir das BUSCH-Netzstrom-Schaltgerät 2087. In diesem Fall bleibt der Piezo-Summer am Ausgang 1 des Computers, während das Netzstrom-Schaltgerät am Ausgang 2 angeschlossen wird. (wie dies bei der Abbildung auf Seite 43 demonstriert wurde). Am Netzstrom-Schaltgerät wird der Rundfunkempfänger angeschlossen.

Das eingegebene Programm „Schaltuhr“ bleibt unverändert bestehen.

Die Abbildung zeigt, daß wir an drei Computer-Ausgängen Anschlußmöglichkeiten für das Programm „Schaltuhr“ haben.

Am Ausgang Nr. 1 die einfache Transistor-Verstärker-Schaltung, mit welcher z. B. ein selbstgebautes Rundfunkgerät oder andere Schaltungen angesteuert werden können, welche mit den BUSCH-Electronic-Studios 2060, 2065, 2070 oder 2075 möglich sind.

Am Ausgang 2 ist das Schwachstrom-Relais 5964 angeschlossen, für zeitabhängige Steuerungen z. B. Modellbahn-Beleuchtungen usw.

Am Ausgang 3 ist das BUSCH-Netzstrom-Schaltgerät 2087 angeschlossen. Der Ausgang 3 wird immer dann eingeschaltet, wenn der Ausgang 1 **oder** Ausgang 2 **oder** beide Ausgänge eingeschaltet sind. Selbstverständlich können wir auch anstelle der dargestellten Variationen drei verschiedener Anschlußmöglichkeiten z. B. 3 Relais 5964 oder 3 Netzstrom-Schaltgeräte 2087 an den Ausgängen anschließen.

### Anwendungs-Beispiel

Während unserer Abwesenheit soll eine 90 Minuten dauernde Rundfunksendung aufgenommen werden. Die maximale Spielzeit einer üblichen Cassette beläuft sich auf 45 Minuten (wenn sie nicht gewendet wird). Wenn wir zwei Cassetten-Recorder besitzen, schließen wir drei Netzstrom-Schaltgeräte 2087 an den Ausgängen 1 bis 3 an. Am Netzstrom-Schaltgerät Ausgang 1 wird der erste Cassetten-Recorder mit 45 Minuten Spielzeit angeschlossen. Am Netzstrom-Schaltgerät des Ausgangs 2 der zweite Cassetten-Recorder. Das für die Aufnahme notwendige Rundfunkgerät wird am Netzstrom-Schaltgerät des Ausgangs 3 angeschlossen.

Die Rundfunksendung soll beispielsweise von 15.00 Uhr bis 16.30 Uhr dauern. Die erste Einschaltzeit ist 15.00 Uhr. Die erste Ausschaltzeit 15.45 Uhr. Die zweite Einschaltzeit ist 15.45 Uhr, zweite Ausschaltzeit 16.30 Uhr. Der Rundfunkempfänger ist während der gesamten Zeit in Betrieb, weil er am Netzschatzgerät des Ausgangs 3 angeschlossen ist.

## Schaltuhr: Das interessiert den Programmierer

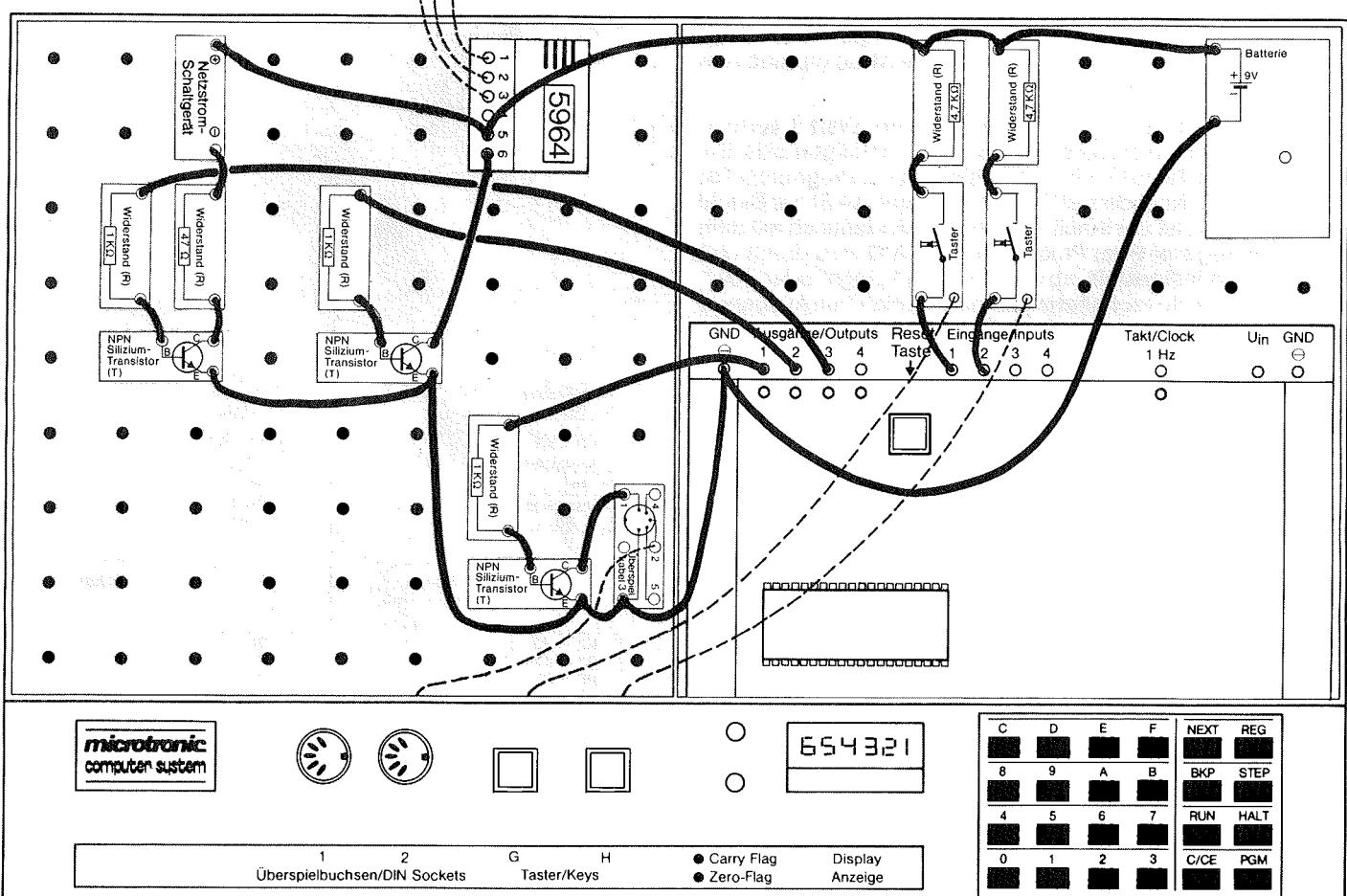
Für das Programm Schaltuhr werden alle zur Verfügung stehenden 16 Arbeits-Register und die 8 Speicher-Register 0 bis 7 verwendet. In den Speicher-Registern 0 bis 7 stehen die Schaltzeiten für den Ausgang 1. In den Arbeits-Registern 0 bis 7 die Schaltzeiten für den Ausgang 2.

Es ergibt sich folgende Register-Belegung:

Register Nr.	Register-Funktionen
0	1-er Minuten
1	10-er Minuten
2	1-er Stunden
3	10-er Stunden
4	1-er Minuten
5	10-er Minuten
6	1-er Stunden
7	10-er Stunden
8	Eingabe-Register (für Tastatur u. Eingänge)
9	Ausgabe-Register (Bitmuster für Ausgänge)
A	1-er Sekunden
B	10-er Sekunden
C	1-er Minuten
D	10-er Minuten
E	1-er Stunden
F	10-er Stunden
Einschaltzeit (Arbeits- bzw. Speicher-Register)	
Ausschaltzeit (Arbeits- bzw. Speicher-Register)	
Uhrzeit (Time-Befehl)	

Nachdem die Sekunden beim Schaltuhr-Betrieb nicht angezeigt werden, kann das Register A für weitere Berechnungen mitverwendet werden.

## Anschluß von Relais usw. an „Schalt-Uhr“



Die Funktionen ergeben sich aus dem Programm-Ablaufplan. In der ersten Spalte bei **START** finden wir die übliche Eingabe-Routine. Folgende Abkürzungen sollen noch erklärt werden:

**EIN 1** = erste Einschaltzeit  
**AUS 1** = erste Ausschaltzeit  
**EIN 2** = zweite Einschaltzeit  
**AUS 2** = zweite Ausschaltzeit  
**S UHR** = Schaltuhr  
**UHR** = Uhrenbetrieb ohne Schaltmöglichkeit

Die Eingabe der Schaltzeiten erfolgt durch die zwei Unter-Programme:

SUB-Routine **S EIN** = Stellen der Einschaltzeit  
SUB-Routine **S AUS** = Stellen der Ausschaltzeit

Im Programm-Teil 3 wird die Uhrzeit ohne Schaltmöglichkeit betrieben. Durch den SHL-Befehl (schiebe nach links) wird der am Eingang 4 vorhandene Takt-Impuls aus dem Register 8 eliminiert, damit kein Programm-Rücksprung zu **START** erfolgt.

Die Schaltuhren-Funktionen finden wir im Programm-Teil 2 **S UHR**. Ein Vergleich der momentanen Uhrzeit mit den eingegebenen Schaltzeiten erfolgt in den Unter-Programmen: SUB-Routine **VEIN** = vergleiche Einschaltzeit, bzw. **VAUS** = vergleiche Ausschaltzeit. In diesen Unter-Programmen wird das Carry-Flag gesetzt (Wert 1) solange eine Schaltzeit nicht mit der Uhrzeit übereinstimmt. Das Carry-Flag wird zurückgesetzt (Wert 0), wenn die momentane Uhrzeit mit der eingegebenen Schaltzeit übereinstimmt. Durch Abfrage des Carry-Flags im Hauptprogramm wird durch die AND- oder OR-Befehle die einzelnen Ausgänge ein- bzw. ausgeschaltet.

Im Programm-Teil 2 **S UHR** wird kontrolliert, ob eine der beiden roten Tasten betätigt wurde. Die Ausgänge werden entsprechend geschaltet. Solange keine der roten Tasten G und H betätigt wird, hat das Register 8 den Wert 0 und der folgende Programm-Teil wird übersprungen.

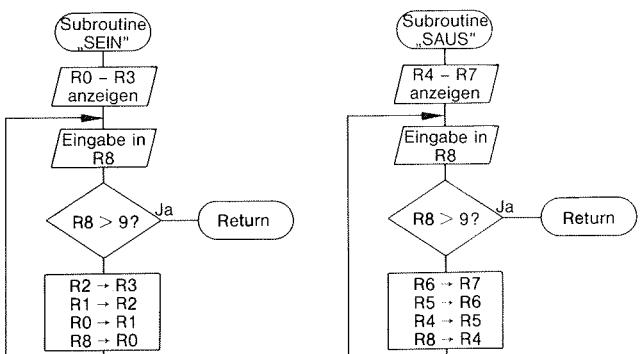
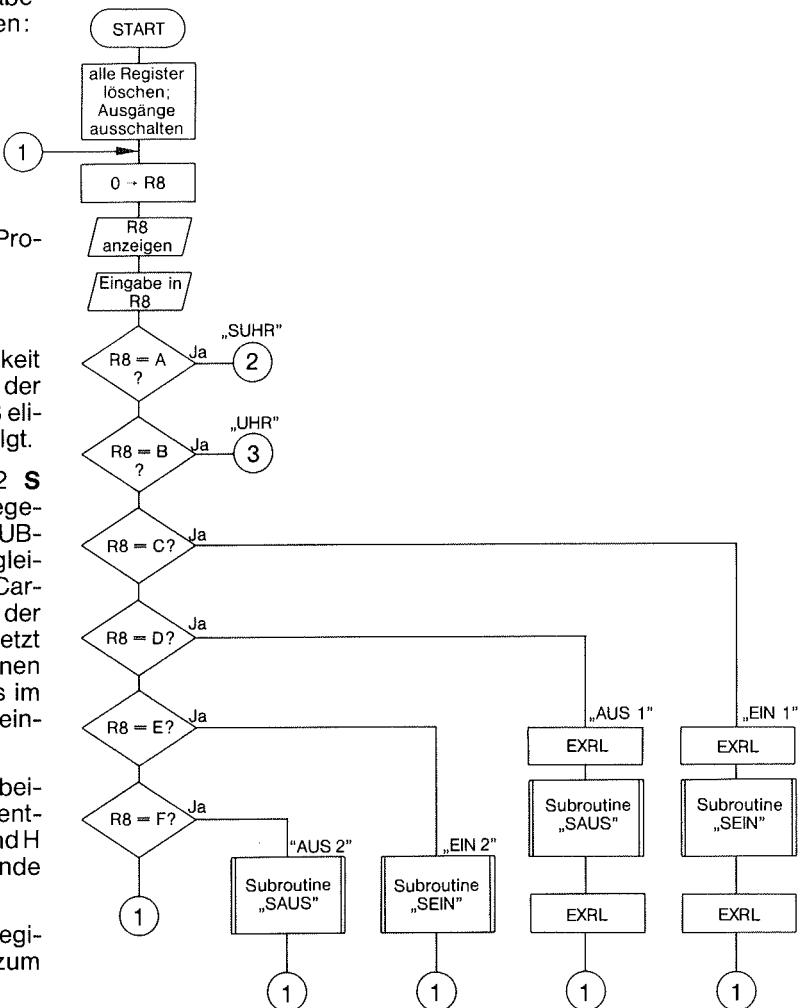
Wurden jedoch beide roten Tasten betätigt, ergibt sich in Register 8 der Wert 3 (dual: 0011) und es erfolgt ein Rücksprung zum Programm-Teil **START**.

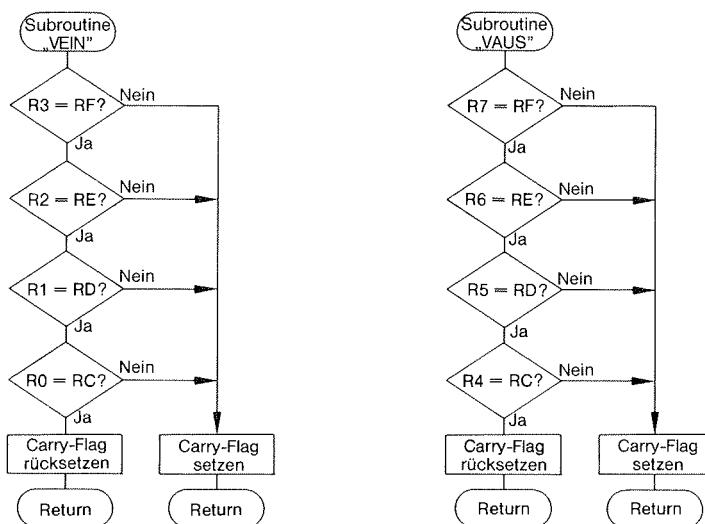
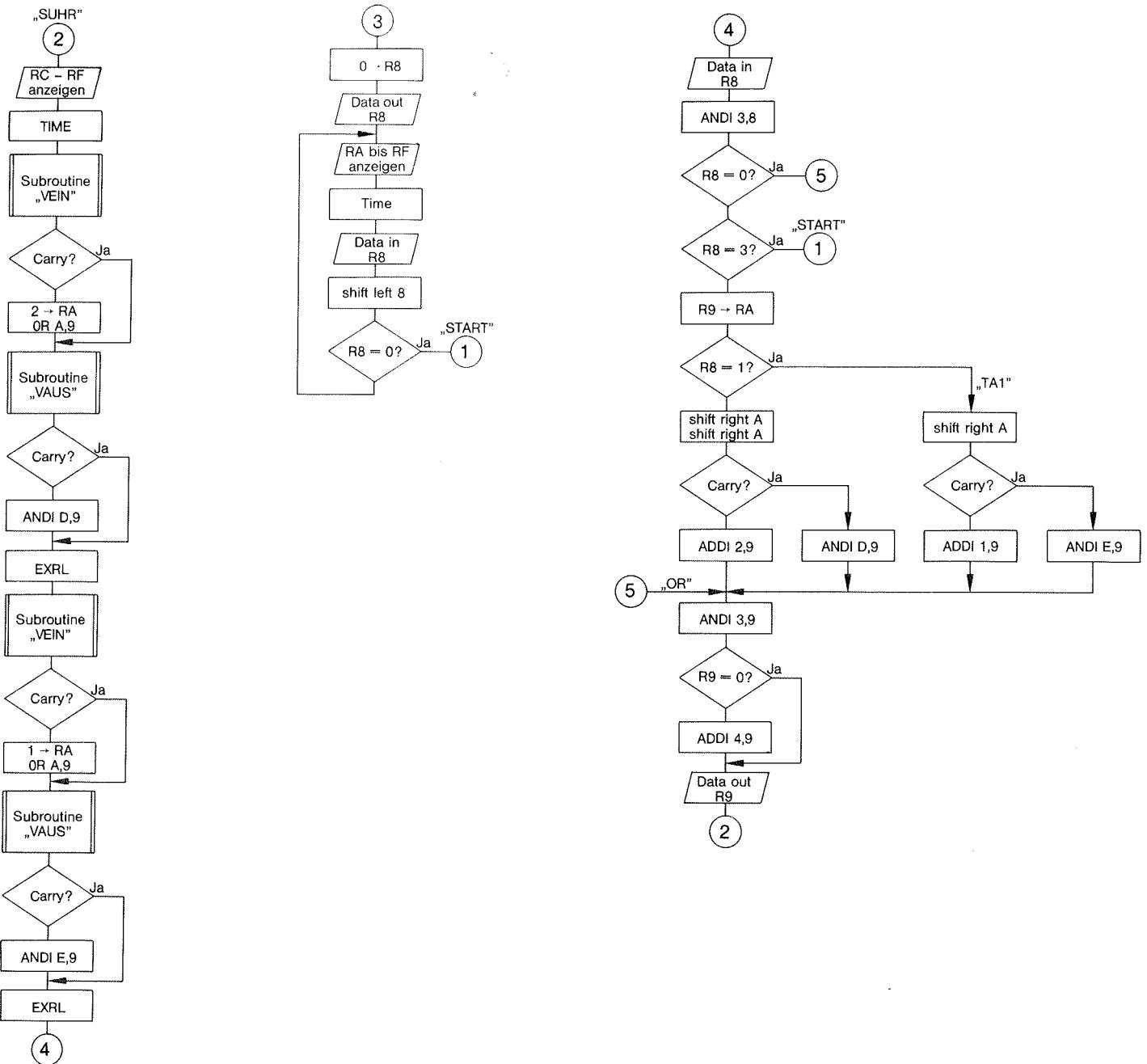
Der nächste Vergleich: „Ist in Register 8 der Wert 1 vorhanden?“ stellt fest, ob die rote Taste G oder H betätigt wurde. Bei Betätigung der Taste G erfolgt ein Sprung zum Programm-Teil **TA G** (Taste G). Nachdem sich zuvor bei Adresse 51 der Befehl MOV 9,A ergibt, ist der Inhalt von Register A identisch mit dem Inhalt von Register 9. Im Programm-Teil **TA G** wird durch den Befehl SHR A festgestellt, ob der Ausgang 1 „high“ oder „low“, d.h. ein- oder ausgeschaltet ist. Bei einem „high“ am Ausgang 1 (dual: 0001) wird durch den SHR-Befehl (schiebe nach rechts) das Carry-Flag gesetzt und durch den Befehl ANDI E, 9 (dual: 1110) der Ausgang abgeschaltet. War jedoch der Ausgang „low“, wird er durch den Befehl ADDI 1,9 (dual: 0001) eingeschaltet.

Das gleiche gilt prinzipiell auch für den Programm-Teil **TA H** (Taste H).

Im Programm-Teil **OR** wird festgestellt, ob einer der beiden Ausgänge Nr. 1 oder 2 „high“, also eingeschaltet ist. Bei eingeschalteten Ausgängen Nr. 1 oder 2 wird auch der Ausgang 3 eingeschaltet.

### Programm-Ablaufplan: Schalt-Uhr





# Der Computer macht Musik

(BUSCH-Electronic-Studio Nr. 2060 oder 2065 oder 2070 erforderlich).

## Eine Mini-Orgel

Nach HALT – NEXT – 00 geben wir das kurze Programm ein:

### Mini-Orgel: Lang-Ton-Version

Adresse	Eingabe-Befehl	Mnemonic
00	<b>F10</b>	DISP 10
01	<b>FF0</b>	KIN 0
02	<b>FE0</b>	DOT 0
03	<b>C00</b>	GOTO 00

Programm-Start: HALT – NEXT – 00 – RUN. Das Display zeigt 0. Jetzt noch die Zahlentaste 0 kurz betätigen. Dann die kleine elektronische Zusatzschaltung gemäß Abbildung aufbauen.

Die Besitzer eines Electronic-Studios 2060 oder 2065 bauen die Schaltung direkt im Computer-Gehäuse auf. Die Besitzer des größeren Electronic-Studios 2070 bauen die Schaltung im Studio-Gehäuse auf und führen die entsprechenden Anschlußkabel zu den Computer-Ausgängen Nr. 1 bis 4. Da sowohl beim Studio-Center 2070 als auch beim Computer Überspielbuchsen vorhanden sind, können die für die Computer-Ausgänge bestimmten Leitungen zunächst auch auf den Baustein „Überspielkabel“ geführt werden. In diesem Fall wird auch beim Computer der Baustein „Überspielkabel“ verwendet und von dort aus die Verbindungsleitungen zu den Computer-Ausgängen hergestellt. Es ist dann darauf zu achten, daß bei beiden Bausteinen „Überspielkabel“ für die Anschlüsse die gleiche Numerierung verwendet wird. Beide Geräte können jetzt durch ein genormtes Überspielkabel miteinander verbunden werden.

Betätigen wir nun am Computer die Zahlentaste Nr. 4 ertönt ein Ton. Bei Eingabe der Zahl 5 ergibt sich ein höherer Ton. Wenn wir die Zahlen- und Buchstabentasten der Reihe nach betätigen, stehen uns 12 Töne zur Verfügung. Bei Betätigung der Taste 0 wird die Tonerzeugung solange unterbrochen, bis wieder eine Zahlen- oder Buchstabentaste betätigt wird.

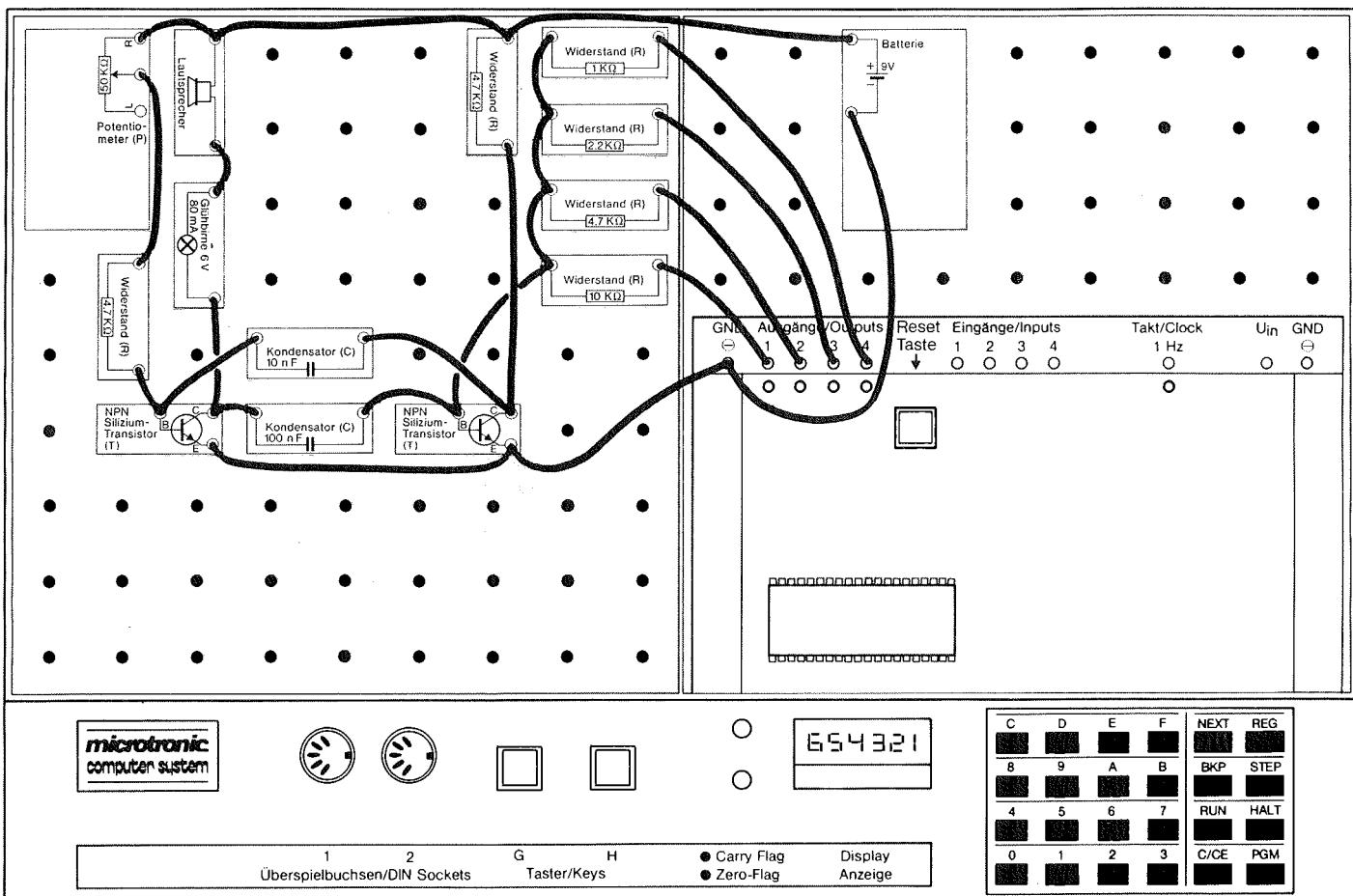
Am Drehknopf des Potentiometers kann die Tonhöhe justiert werden. Dies ist erforderlich, um eine tonleiterähnliche Anordnung zu erhalten. Mit den Tasten 1 bis 3 ergeben sich keine „brauchbaren“ Töne.

Unsere Mini-Orgel hat den Nachteil, daß in der jetzt programmierten Version Dauertöne produziert werden. Wir können die Programmierung ändern, damit durch Tastendruck nur einzelne Töne erzeugt werden. Nach HALT – NEXT – 00 geben wir folgendes Programm ein:

### Mini-Orgel: Kurz-Ton-Version

Adresse	Eingabe-Befehl	Mnemonic
00	<b>F10</b>	DISP 1,0
01	<b>FF0</b>	KIN 0
02	<b>FE0</b>	DOT 0
03	<b>511</b>	ADDI 1,1
04	<b>E06</b>	BRZ 06
05	<b>C03</b>	GOTO 03
06	<b>FE1</b>	DOT 1
07	<b>C00</b>	GOTO 00

### Mini-Orgel aus Bauelementen der Electronic-Studios 2060 oder 2065



Programm-Start: HALT – NEXT – 00 – RUN. Das Display zeigt 0, der Computer erzeugt einen Dauerton, bis die Taste 0 betätigt wird. Werden jetzt die verschiedenen Zahlen- und Buchstaben-tasten gedrückt, ergeben sich Töne, die sich nach kurzer Zeit automatisch abstellen. Wir können die Tonfolge weiterverkürzen, wenn wir den Befehl ADDI auf Adresse 03 von 511 abändern in 521 (oder 541, oder 581).

### Das interessiert den Programmierer

Das zuerst eingegebene Programm haben wir bereits auf Seite 16 im ersten Teil des Anleitungsbuches kennengelernt (Dual-System). Je nach Tastenbetätigung wird ein dualer Wert an die Ausgänge gebracht. Die entsprechenden LED's an den Ausgängen leuchten. Ausgänge mit leuchtenden LED's sind „high“, d. h. daß dort eine geringe Spannung von ca. 3 bis 5 Volt vorhanden ist.

In der elektronischen Zusatzaufschaltung sind die Transistoren T1 und T2 zu einer astabilen Kippstufe geschaltet. An die Basis des Transistors T2 kommt über die Computer-Ausgänge Nr. 1 bis 4 eine Basisspannung, welche durch die verschiedenen Widerstände verändert wird. Hierdurch ergibt sich eine veränderbare Tonfrequenz.

Sind alle Ausgänge „low“ (LED's abgeschaltet, Display zeigt 0), fehlt dem Transistor T2 die Basisspannung, folglich können keine Schwingungen erzeugt werden. Wird die Taste 1 betätigt (LED am Ausgang 1 leuchtet), ergibt sich durch den 10 kΩ-Widerstand (beim Studio 2070 durch 22 kΩ), eine sehr geringe Basisspannung, welche für die Erzeugung von Tonschwingungen noch nicht ausreicht. Durch Betätigung der verschiedenen Zahlen- und Buchstabentasten ergeben sich jedoch verschiedene Widerstandskombinationen, wodurch Schwingungen in den verschiedenen Tonfrequenzbereichen entstehen.

Das Klangvolumen kann mit dem Verstärker-Baustein des preiswerten Electronic-Studio-Zusatzaufschaltungs „IC-Verstärker-Technik 2072“ verbessert werden. Aus dem Anleitungsbuch 2072 ist es ersichtlich, wie der Verstärker-Baustein bei solchen Schaltungen angeordnet wird.

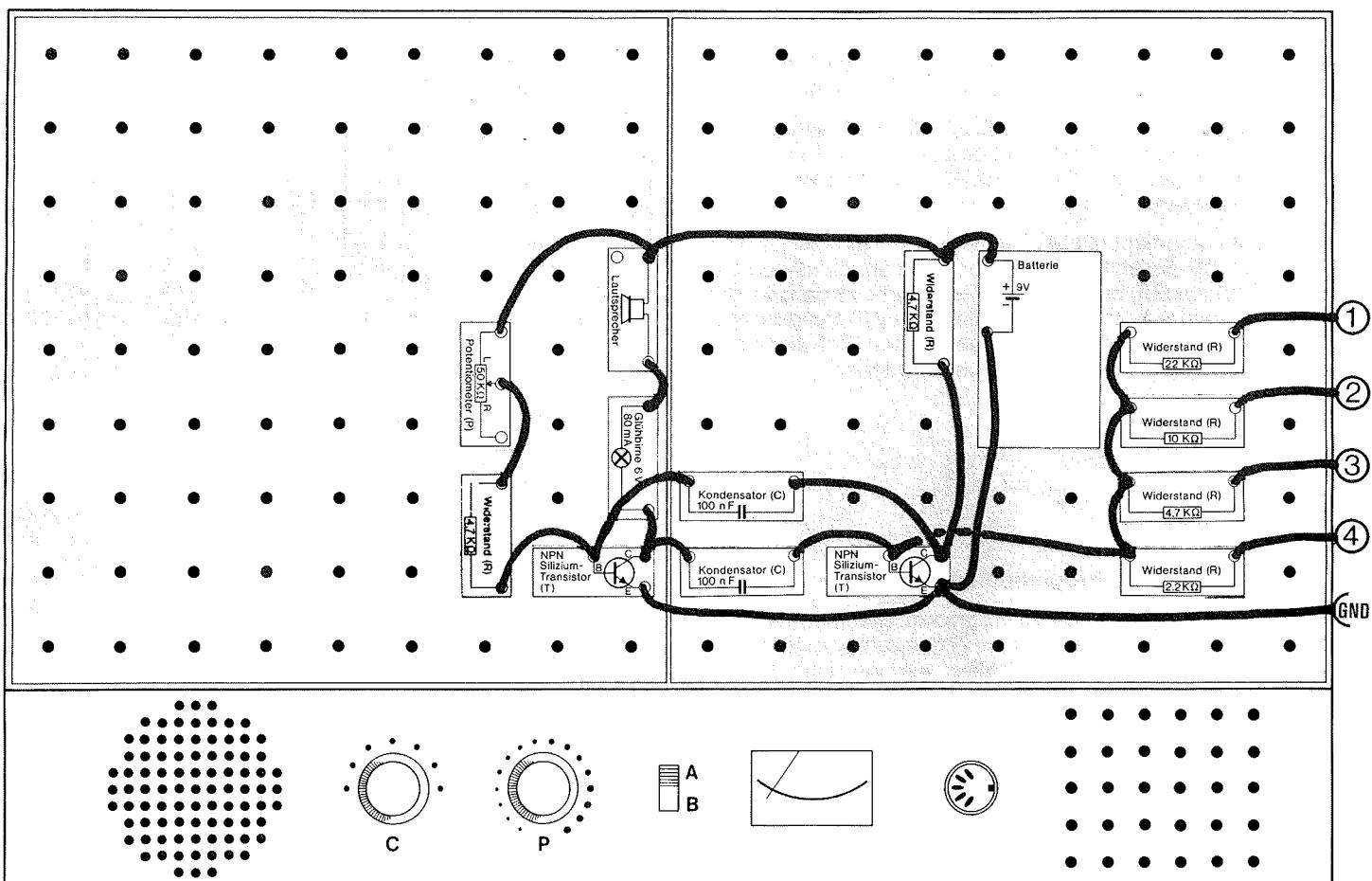
Es wurde bereits erklärt, daß für elektronische Zusatzaufschaltungen die an den Computer-Ausgängen zur Verfügung stehenden Spannungen verstärkt werden müssen, d. h. daß eine zusätzliche Spannungsquelle (Batterie oder BUSCH-Netzgerät 2059) erforderlich ist. Bei allen derartigen Schaltungen muß der Minus-Pol der Batterie auch mit dem Minus-Pol des Computers verbunden werden. Der Minus-Pol des Computers wird als „ground“ (GND) bezeichnet. Auf der Computer-Platine erkennen wir sowohl links als auch rechts eine GND-Anschlußbuchse. Für elektronische Schaltungen ist grundsätzlich die linke GND-Buchse zu verwenden. Die rechte GND-Buchse ist lediglich für den Batterie-Anschluß zur Spannungsversorgung des RAMs bei Netzstromunterbrechung zu benutzen.

Der Computer hat eine Buchse mit der Bezeichnung  $U_{in}$ . Diese darf niemals mit dem Minus-Pol der Batterie verbunden werden (Kurzschlußgefahr!).

Bei der zweiten Programm-Version (Mini-Orgel mit Kurzton-Erzeugung) ist der ADDI-Befehl auf Adresse 03 für die Tonlänge verantwortlich. In Register 1 wird laufend der konstante Wert 1 hinzuaddiert (automatischer Zähler), bis sich ein Übertrag ergibt, welches das Zero-Flag auslöst. Wird durch Änderung des ADDI-Befehls, z. B. der konstante Wert 2 addiert, wird der Zählvorgang beschleunigt, die Tonerzeugung wird verkürzt.

Die Versuchsanordnung bleibt auch für das folgende Experiment bestehen.

**Mini-Orgel aufgebaut mit dem Electronic-Studio 2070.** Die vier Anschlußleitungen führen zu den Computer-Ausgängen.



## Der komponierende Computer (Melodien-Generator)

Der Computer soll selbsttätig Melodien komponieren mit langen und kurzen, hohen und tiefen Tönen. Ein interessantes Experiment, für welches der gleiche Schaltungsaufbau wie bei der vorangegangenen Mini-Orgel verwendet wird.

Es ist lediglich eine neue Programmierung erforderlich. Damit uns beim Programmieren die Tonerzeugung des Computers nicht stört, wird eine Verbindungsleitung zur Batterie abgeklemmt. Nach HALT – NEXT – 00 folgendes Programm eingegeben:

### Programm: Der komponierende Computer

Adresse	Eingabe-Befehl	Mnemonic
00	<b>F08</b>	CLEAR
01	<b>FE0</b>	DOT 0
02	<b>F60</b>	DISP 6,0
03	<b>FF0</b>	KIN 0
04	<b>F05</b>	RND
05	<b>531</b>	ADDI 3,1
06	<b>40D</b>	ADD 0,D
07	<b>E0D</b>	BRZ 0D
08	<b>41D</b>	ADD 1,D
09	<b>E04</b>	BRZ 04
0A	<b>93D</b>	CMPI 3,D
0D	<b>D0D</b>	BRC 0D
0C	<b>57D</b>	ADDI 7,D
0D	<b>FED</b>	DOT 0
0E	<b>71E</b>	SUBI 1,E
0F	<b>E04</b>	BRZ 04
10	<b>C0E</b>	GOTO 0E

Programm-Start mit HALT – NEXT – 00 – RUN. Das Display zeigt: 000000. Die Verbindungsleitung zur Batterie wieder anschließen. Sobald eine Zahlen- oder Buchstabetaste betätigt wird, komponiert der Computer zufällige Melodienfolgen.

Das Programm wird mit HALT – NEXT – 00 – RUN gestoppt. Der Melodien-Generator arbeitet weiter, sobald eine Zahl oder ein Buchstabe eingegeben wird. Es ergeben sich immer neue Melodien-Variationen.

Weitere Variationsmöglichkeiten ergeben sich durch Änderung des DISP-Befehls auf Adresse 02. Wir ändern die dort stehende Eingabe F60 in F10. NEXT-Taste nicht vergessen, dann neuer Programm-Start. Jetzt ergeben sich völlig andere Melodienfolgen. Wir können den DISP-Befehl mehrfach ändern und es ergeben sich immer wieder neue Variationen.

### Das interessiert den Programmierer:

Ein sehr einfach übersehbares Programm. Durch den RND-Befehl wird in die Register D bis F eine Zufallszahl geladen und um dem Zufall noch etwas nachzuhelfen, wird der nach dem Programm-Start eingegebene Zahlenwert im Register D dazugeaddiert.

Der Wert in Register D ist für die Tonhöhe verantwortlich. Durch Abwärtszählen der Zufallszahl in Register E wird die Tonlänge erzeugt.

Die Versuchsanordnung bleibt auch für das folgende Experiment bestehen.

## Computer steuert elektronische Spieluhr

Bei den vorangegangenen Experimenten haben wir durch Tastenbetätigung Töne erzeugt, bzw. den Computer mit Hilfe des Zufallsgenerators veranlaßt, selbsttätig Melodien zu komponieren.

Mit diesem Programm wollen wir dem Computer beibringen „richtige Melodien“ automatisch zu spielen. Damit sich einwandfreie Tonfolgen ergeben, ist der Schaltungsaufbau mit dem Electronic-Studio 2070 (siehe Seite 53) erforderlich. Wer nur das kleinere Electronic-Studio 2060 besitzt, kann sich über den BUSCH-Ersatzteildienst die beiden noch fehlenden Baulemente kommen lassen:

Best. Nr. 20669 1 Scheibenkondensator 100 nF

Best. Nr. 20689 1 Widerstand 22 K Ω

(Bitte DM 3,50 inkl. Porto und Verpackung auf Postscheck-Konto Karlsruhe Nr. 57 614-751 Busch GmbH & Co. KG überweisen und auf dem Empfänger-Ab schnitt die Bestellnummern angeben).

Mit folgender Programmierung spielt der Computer mit der angeschlossenen elektronischen Schaltung nacheinander die drei bekannten Melodien: „Sentimental Journey – Hänschen Klein – Du, Du liegst mir am Herzen“.

Bevor wir die Programmierung nach HALT – NEXT – 00 beginnen, wird die elektronische Schaltung durch Abklemmen des Batteriekabels außer Funktion gesetzt.

### Programm: Spiel-Uhr

Adr. Nr.	Eingabe-Befehl	Mnemonic
00	<b>F02</b>	DISOUT
01	<b>C0B</b>	GOTO 0B

#### (Unterprogramm)

02	<b>FE0</b>	DOT 0
03	<b>511</b>	ADDI 1,1
04	<b>E06</b>	BRZ 06
05	<b>C03</b>	GOTO 03
06	<b>902</b>	CMPI 0,2
07	<b>E09</b>	BRZ 09
08	<b>F07</b>	RET
09	<b>FE1</b>	DOT 1
0A	<b>F07</b>	RET

#### („Sentimental Journey“)

0B	<b>112</b>	MOVI 1,2
0C	<b>180</b>	MOVI 8,0
0D	<b>B02</b>	CALL 02
0E	<b>140</b>	MOVI 4,0
0F	<b>B02</b>	CALL 02
10	<b>180</b>	MOVI 8,0
11	<b>B02</b>	CALL 02
12	<b>140</b>	MOVI 4,0
13	<b>B02</b>	CALL 02
14	<b>180</b>	MOVI 8,0
15	<b>B02</b>	CALL 02
16	<b>140</b>	MOVI 4,0
17	<b>B02</b>	CALL 02
18	<b>100</b>	MOVI 0,0
19	<b>B02</b>	CALL 02
1A	<b>180</b>	MOVI 8,0
1B	<b>B02</b>	CALL 02

1C	<b>140</b>	MOVI 4,0
1D	<b>B02</b>	CALL 02
1E	<b>180</b>	MOVI 8,0
1F	<b>B02</b>	CALL 02
20	<b>190</b>	MOVI 9,0
21	<b>B02</b>	CALL 02
22	<b>180</b>	MOVI 8,0
23	<b>B02</b>	CALL 02
24	<b>170</b>	MOVI 7,0
25	<b>B02</b>	CALL 02
26	<b>160</b>	MOVI 6,0
27	<b>B02</b>	CALL 02
28	<b>100</b>	MOVI 0,0
29	<b>B02</b>	CALL 02
2A	<b>180</b>	MOVI 8,0
2B	<b>B02</b>	CALL 02
2C	<b>140</b>	MOVI 4,0
2D	<b>B02</b>	CALL 02
2E	<b>180</b>	MOVI 8,0
2F	<b>B02</b>	CALL 02
30	<b>140</b>	MOVI 4,0
31	<b>B02</b>	CALL 02
32	<b>170</b>	MOVI 7,0
33	<b>B02</b>	CALL 02
34	<b>140</b>	MOVI 4,0
35	<b>B02</b>	CALL 02
36	<b>170</b>	MOVI 7,0
37	<b>B02</b>	CALL 02
38	<b>140</b>	MOVI 4,0
39	<b>B02</b>	CALL 02
3A	<b>180</b>	MOVI 8,0
3B	<b>B02</b>	CALL 02

3C	<b>140</b>	MOVI 4,0
3D	<b>B02</b>	CALL 02
3E	<b>100</b>	MOVI 0,0
3F	<b>B02</b>	CALL 02
40	<b>180</b>	MOVI 8,0
41	<b>B02</b>	CALL 02
42	<b>190</b>	MOVI 9,0
43	<b>B02</b>	CALL 02
44	<b>180</b>	MOVI 8,0
45	<b>B02</b>	CALL 02
46	<b>170</b>	MOVI 7,0
47	<b>B02</b>	CALL 02
48	<b>160</b>	MOVI 6,0
49	<b>B02</b>	CALL 02
4A	<b>140</b>	MOVI 4,0
4B	<b>B02</b>	CALL 02
4C	<b>100</b>	MOVI 0,0
4D	<b>B02</b>	CALL 02

71	<b>1A0</b>	MOVI A,0
72	<b>B02</b>	CALL 02
73	<b>160</b>	MOVI 6,0
74	<b>B02</b>	CALL 02
75	<b>160</b>	MOVI 6,0
76	<b>B02</b>	CALL 02
77	<b>140</b>	MOVI 4,0
78	<b>B02</b>	CALL 02
79	<b>180</b>	MOVI 8,0
7A	<b>B02</b>	CALL 02
7B	<b>1C0</b>	MOVI C,0
7C	<b>B02</b>	CALL 02
7D	<b>1C0</b>	MOVI C,0
7E	<b>B02</b>	CALL 02
7F	<b>140</b>	MOVI 4,0
80	<b>B02</b>	CALL 02
81	<b>100</b>	MOVI 0,0
82	<b>B02</b>	CALL 02

A6	<b>102</b>	MOVI 0,2
A7	<b>B02</b>	CALL 02
A8	<b>112</b>	MOVI 1,2
A9	<b>B02</b>	CALL 02
AA	<b>102</b>	MOVI 0,2
AB	<b>B02</b>	CALL 02
AC	<b>112</b>	MOVI 1,2
AD	<b>B02</b>	CALL 02
AE	<b>102</b>	MOVI 0,2
AF	<b>B02</b>	CALL 02
B0	<b>B02</b>	CALL 02
B1	<b>140</b>	MOVI 4,0
B2	<b>B02</b>	CALL 02
B3	<b>160</b>	MOVI 6,0
B4	<b>112</b>	MOVI 1,2
B5	<b>B02</b>	CALL 02
B6	<b>190</b>	MOVI 9,0
B7	<b>B02</b>	CALL 02
B8	<b>B02</b>	CALL 02
B9	<b>170</b>	MOVI 7,0
BA	<b>B02</b>	CALL 02
BB	<b>102</b>	MOVI 0,2

BC	<b>B02</b>	CALL 02
BD	<b>140</b>	MOVI 4,0
BE	<b>B02</b>	CALL 02
BF	<b>160</b>	MOVI 6,0
C0	<b>B02</b>	CALL 02
C1	<b>170</b>	MOVI 7,0
C2	<b>B02</b>	CALL 02
C3	<b>1C0</b>	MOVI C,0
C4	<b>B02</b>	CALL 02
C5	<b>190</b>	MOVI 9,0
C6	<b>B02</b>	CALL 02
C7	<b>170</b>	MOVI 7,0
C8	<b>B02</b>	CALL 02
C9	<b>160</b>	MOVI 6,0
CA	<b>B02</b>	CALL 02
CB	<b>112</b>	MOVI 1,2
CC	<b>140</b>	MOVI 4,0
CD	<b>B02</b>	CALL 02
CE	<b>B02</b>	CALL 02
CF	<b>B02</b>	CALL 02
D0	<b>100</b>	MOVI 0,0
D1	<b>B02</b>	CALL 02

#### („Hänschen Klein“)

4E	<b>102</b>	MOVI 0,2
4F	<b>1C0</b>	MOVI C,0
50	<b>B02</b>	CALL 02
51	<b>180</b>	MOVI 8,0
52	<b>B02</b>	CALL 02
53	<b>180</b>	MOVI 8,0
54	<b>B02</b>	CALL 02
55	<b>1A0</b>	MOVI A,0
56	<b>B02</b>	CALL 02
57	<b>160</b>	MOVI 6,0
58	<b>B02</b>	CALL 02
59	<b>160</b>	MOVI 6,0
5A	<b>B02</b>	CALL 02
5B	<b>140</b>	MOVI 4,0
5C	<b>B02</b>	CALL 02
5D	<b>160</b>	MOVI 6,0
5E	<b>B02</b>	CALL 02
5F	<b>180</b>	MOVI 8,0
60	<b>B02</b>	CALL 02
61	<b>1A0</b>	MOVI A,0
62	<b>B02</b>	CALL 02
63	<b>1C0</b>	MOVI C,0
64	<b>B02</b>	CALL 02
65	<b>1C0</b>	MOVI C,0
66	<b>B02</b>	CALL 02
67	<b>1C0</b>	MOVI C,0
68	<b>B02</b>	CALL 02
69	<b>100</b>	MOVI 0,0
6A	<b>B02</b>	CALL 02
6B	<b>1C0</b>	MOVI C,0
6C	<b>B02</b>	CALL 02
6D	<b>180</b>	MOVI 8,0
6E	<b>B02</b>	CALL 02
6F	<b>180</b>	MOVI 8,0
70	<b>B02</b>	CALL 02

#### („Du liegst mir am Herzen“)

83	<b>112</b>	MOVI 1,2
84	<b>160</b>	MOVI 6,0
85	<b>B02</b>	CALL 02
86	<b>102</b>	MOVI 0,2
87	<b>B02</b>	CALL 02
88	<b>B02</b>	CALL 02
89	<b>102</b>	MOVI 0,2
8A	<b>B02</b>	CALL 02
8B	<b>B02</b>	CALL 02
8C	<b>140</b>	MOVI 4,0
8D	<b>B02</b>	CALL 02
8E	<b>160</b>	MOVI 6,0
8F	<b>B02</b>	CALL 02
90	<b>112</b>	MOVI 1,2
91	<b>190</b>	MOVI 9,0
92	<b>B02</b>	CALL 02
93	<b>B02</b>	CALL 02
94	<b>170</b>	MOVI 7,0
95	<b>B02</b>	CALL 02
96	<b>140</b>	MOVI 4,0
97	<b>B02</b>	CALL 02
98	<b>102</b>	MOVI 0,2
99	<b>B02</b>	CALL 02
9A	<b>112</b>	MOVI 1,2
9B	<b>B02</b>	CALL 02
9C	<b>102</b>	MOVI 0,2
9D	<b>B02</b>	CALL 02
9E	<b>B02</b>	CALL 02
9F	<b>190</b>	MOVI 9,0
A0	<b>B02</b>	CALL 02
A1	<b>170</b>	MOVI 7,0
A2	<b>B02</b>	CALL 02
A3	<b>112</b>	MOVI 1,2
A4	<b>160</b>	MOVI 6,0
A5	<b>B02</b>	CALL 02

Batteriekabel zur elektronischen Schaltung wieder anklemmen. Programm-Start mit HALT – NEXT – 00 – RUN.

Sofort beginnt der Computer die 3 Melodien nacheinander zu spielen, wobei er solange wiederholt, bis die grüne RESET-Taste auf der Computer-Platine betätigt wird. Neuer Programm-Start wie üblich mit HALT – NEXT – 00 – RUN.

Mit dem Drehknopf des Potentiometers kann die Tonhöhe etwas nachgeregelt werden, falls die Melodien nicht in der richtigen Tonlage gespielt werden.

Falls sich die Besitzer der Electronic-Studios 2060, bzw. 2065 die beiden vorerwähnten Bauelemente nicht kommen lassen, ist dieser Versuch trotzdem durchführbar, es müssen jedoch dann an einigen Stellen falsche Tonfolgen hingenommen werden.

#### Das interessiert den Programmierer:

Durch hexadezimale Eingaben werden duale Werte an die Ausgänge gebracht, wodurch sich dort die erwünschten „high“ und „low“ Zustände ergeben. Je nach dem, welche Ausgänge und wieviele der Ausgänge „high“ sind, werden die verschiedenen Tonlagen gespielt.

Um den Computer zu veranlassen, bestimmte Melodien zu spielen, müssen die erforderlichen Töne in der richtigen Reihenfolge durch die entsprechenden dualen Werte an den Ausgängen erzeugt werden. Wir müssen wissen, welche Widerstands-Kombination unserer Schaltung den gewünschten Ton erzeugt. Dementsprechend ergibt sich die Programmierung der Tonfolgen.

Nachdem die 3 Melodien aus einer ganzen Menge einzelner Töne bestehen, ist ein längeres Programm erforderlich, weil jeder einzelne Ton programmiert werden muß.

Trotzdem ergibt sich auch ohne Programm-Ablaufplan eine einfache und übersichtliche Darstellung, weil das gesamte „Ton-Programm“ (Tonhöhe) im Register 0 und (Tonlänge) im Register 2 durchgeführt wird.

## Akustische Effekte für Computer-Spiele:

Die MOVI-Befehle bringen die hexadezimalen Werte für die duale Darstellung an den Ausgängen in das Register 0. Durch die CALL-Befehle erfolgt ein Sprung ins Unter-Programm (ab Adresse 02 bis 0A), wodurch die entsprechenden Ausgänge für kurze Zeit aktiviert werden.

Die MOVI-Befehle sorgen auch für die Tonlängen, indem entweder der Wert 0 (kurze Töne) oder der Wert 1 (lange Töne) in das Register 2 gebracht wird. Hat das Register 2 den Inhalt 1 ergibt sich ein Dauerton, welcher sich erst durch den nächst folgenden Ton ändert. Ist in Register 2 der Wert 0, entsteht ein kurzer Ton, d. h. zwischen zwei Tönen ergibt sich eine kurze Pause. Die Melodie „Sentimental Journey“ beginnt bei Adresse OB. Der Befehl MOVI 1,2 bringt den Wert 1 in das Register 2, d. h. daß alle folgenden Ton-Befehle Dauertöne erzeugen. Mit dem folgenden Befehl MOVI 8,0 wird der Wert 8 in das Register 0 gebracht (dual: 1000), d. h. der Ausgang 4 wird „high“ und der entsprechende Ton wird durch die elektronische Schaltung erzeugt. Nach jedem MOVI-Befehl folgt ein CALL-Befehl. Durch dieses ständige Hin- und Herspringen vom „Melodien-Programm“ zum „Unter-Programm“ ergibt sich die Melodienfolge.

Die Melodie „Hänschen klein“ ergibt sich ab Adresse 4E, wobei dort durch den Befehl MOVI 0,2 der Wert 0 in Register 2 gebracht wird, so daß alle folgenden Töne „kurz gespielt“ werden.

Bei der Melodie „Du, Du liegst mir am Herzen“ werden wechselweise die Register 0 und 2 angesprochen, weil für diese Melodie lange und kurze Töne erforderlich sind.

Es sollte uns nicht schwerfallen, den Computer mit einer Melodie, die uns besonders gut gefällt, selbst zu programmieren.

Die im Handel angebotenen „Computer-Spiele“ werden oftmals wegen ihrer akustischen Effekte gekauft. Manches Spiel kann durch eine akustische Darstellung des erreichten Spielstandes interessanter gestaltet werden.

Das Spiel „Die Mondlandung“ (Seite 23, Anleitungsbuch 1. Teil) wird durch akustische Signale am Spielende erheblich effektvoller.

Probieren wir es einmal aus!

Wir geben nochmals das Programm Mondlandung ein. Bis auf wenige Befehle, kann das Programm von Seite 24 übernommen werden:

Bei Adresse 56 (FE6) ist die Änderung in CA5 erforderlich. Bei Adresse 6D (1FF) wird geändert in C99. Bei Adresse 7F (C6D) wird durch C90 und Adresse 84 (C6D) durch C9F ersetzt.

Dann wird das Programm ab Adresse 90 ergänzt. Wir nehmen die zusätzliche Neuprogrammierung nach HALT – NEXT – 90 vor:

### Programmergänzung: Akustische Effekte für „Mondlandung“

Adresse	Eingabe Befehl	Mnemonic	Erklärungen
90	<b>11D</b>	MOVI 1,D	Tonerzeugung für „Mondfahre zerschellt“
91	<b>51E</b>	ADDI 1,E	
92	<b>FBD</b>	ADC D	
93	<b>E73</b>	BRZ 73	
94	<b>FEE</b>	DOT E	
95	<b>0EF</b>	MOV E,F	
96	<b>F8F</b>	INV F	
97	<b>FEF</b>	DOT F	
98	<b>C91</b>	GOTO 91	
99	<b>11D</b>	MOVI 1,D	Tonerzeugung für „Treibstoff verbraucht“
9A	<b>71E</b>	SUBI 1,E	
9B	<b>FBD</b>	ADC D	
9C	<b>E73</b>	BRZ 73	
9D	<b>FEE</b>	DOT E	
9E	<b>C9A</b>	GOTO 9A	
9F	<b>11D</b>	MOVI 1,D	Tonerzeugung für „Mondfahre verschwindet“
A0	<b>51E</b>	ADDI 1,E	
A1	<b>FBD</b>	ADC D	
A2	<b>E73</b>	BRZ 73	
A3	<b>FEE</b>	DOT E	
A4	<b>CA0</b>	GOTO A0	
A5	<b>10D</b>	MOVI 0,D	Tonerzeugung für „Weiche Mondlandung“
A6	<b>14E</b>	MOVI 4,E	
A7	<b>FEE</b>	DOT E	
A8	<b>1AE</b>	MOVI A,E	
A9	<b>FEE</b>	DOT E	
AA	<b>54D</b>	ADDI 4,D	
AB	<b>FBC</b>	ADC D	
AC	<b>DAE</b>	BRC AE	
AD	<b>CA6</b>	GOTO A6	
AE	<b>10E</b>	MOVI 0,E	
AF	<b>FEE</b>	DOT E	
B0	<b>C57</b>	GOTO 57	

## Der Computer zählt Personen und Gegenstände

(BUSCH-Electronic-Studio 2065 oder 2070 erforderlich)

Die tonerzeugende Schaltung aus den vorangegangenen Versuchen wird auch für dieses Experiment unverändert übernommen.

Nach HALT – NEXT – 00 – RUN wird das Programm in gleicher Weise in Betrieb genommen, wie auf Seite 23 des Anleitungsbuches 1. Teil beschrieben. Am Spielende ergeben sich jetzt zusätzliche Akustik-Effekte, welche durch entsprechende Tonfolge dem Spielergebnis angepaßt sind:

**Weiche Landung der Fähre** = melodische Tonfolge aus hohen und tiefen Tönen.

**Fähre ist auf der Mondoberfläche zerschellt** = gleichzeitig ansteigende und abfallende Melodienfolge.

**Treibstoff-Vorrat ist verbraucht** = mehrmals abfallende Tonfolge.

**Fähre verschwindet im All** = schnell ansteigende Tonfolge.

Das Spiel „Mondlandung“ ist jetzt perfekt. Das kurze Zusatz-Programm zeigt, wie man solche Spiele durch akustische Effekte noch interessanter machen kann. Die einzelnen Ton-Programm-Teile können auch für andere Spiel-Programme nachträglich übernommen werden. (Befehlsänderungen für die Sprünge zum Unterprogramm beachten!)

Wo Licht ist, ist auch Schatten! Durch diesen Effekt kann man mit Lichtschranken Zähleffekte erreichen, indem ein beleuchteter Fotowiderstand den Wechsel zwischen hell und dunkel durch Änderung der Spannung an den Computer-Eingang weitergibt.

Wir bauen die elektronische Schaltung gemäß Abbildung auf. Der Fotowiderstand sollte vom Tageslicht oder vom Licht einer Lampe beleuchtet sein. Durch langsames Drehen am Potentiometerknopf wird die Schaltung so einjustiert, daß die Glühlampe gerade nicht mehr leuchtet. Wird der Fotowiderstand, z. B. durch den Schatten der darübergehaltenen Hand etwas abgedunkelt, muß das Glühlämpchen leuchten.

Nachdem die elektronische Schaltung am Computer angeschlossen wurde, nehmen wir die Programmierung nach HALT – NEXT – 00 vor:

### Programm: Lichtschranken gesteuerter Zähler

Adresse	Eingabe-Befehl	Mnemonic	Erklärungen	
00	<b>F08</b>	CLEAR	Alle Register löschen	
01	<b>F21</b>	DISP 2,1	Register 1 und 2 anzeigen	
02	<b>FD0</b>	DIN 0	Eingangs-Wert in Reg. 0 speichern	
03	<b>330</b>	ANDI 3,0	Wert an Eingang 3 u. 4 löschen	Warte-schleife
04	<b>803</b>	CMP 0,3	Reg. 0 = Reg 3?	
05	<b>E02</b>	BRZ 02	Ja, dann Sprung nach 02	
06	<b>003</b>	MOV 0,3	Reg 0 → Reg. 3	
07	<b>910</b>	CMPI 1,0	Reg 0 = 1?	
08	<b>E02</b>	BRZ 02	Ja, Sprung nach 02	
09	<b>D00</b>	BRC 00	Größer als 1, dann Sprung nach 00	
0A	<b>511</b>	ADDI 1,1	Dezimales Zählprogramm in Register 1 und 2	
0B	<b>991</b>	CMPI 9,1	ähnlich wie auf Seite 25 (ab Adresse 08 beschrieben)	
0C	<b>D0E</b>	BRC 0E		
0D	<b>C02</b>	GOTO 02		
0E	<b>561</b>	ADDI 6,1		
0F	<b>512</b>	ADDI 1,2		
10	<b>992</b>	CMPI 9,2		
11	<b>D00</b>	BRC 00		
12	<b>C02</b>	GOTO 02		

Programm-Start: HALT – NEXT – 00 – RUN. Das Display zeigt 00.

Sobald wir (z. B. mit der Hand) einen Schatten auf den Fotowiderstand bringen, wird dieser Wechsel hell – dunkel gezählt. Das Display zeigt 01. (bzw. wie oft ein Schatten vom Fotowiderstand registriert wurde). Mit der roten Taste H kann der Zähler wieder auf 0 zurückgesetzt werden.

Wenn das Verbindungskabel verlängert wird, kann der Fotowiderstand beispielsweise an einer Tür angebracht werden. Der Computer zählt, wieviel Personen durch diese Tür einen Raum betreten haben.

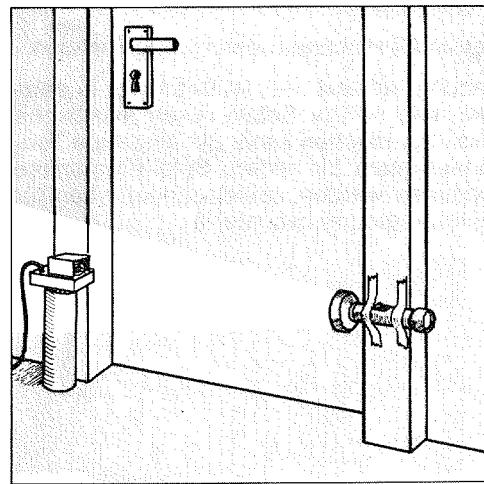
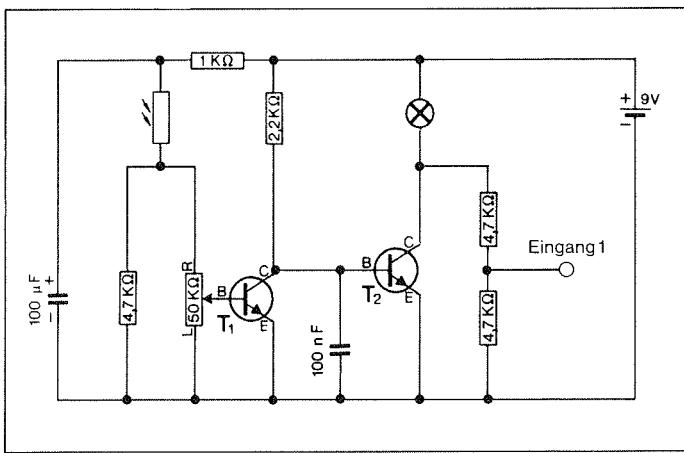
Mit einem gebündelten Lichtstrahl können Gegenstände, z. B. auf einem Fließband gezählt werden. Die Schaltung ist auch als Rundenzähler bei Modellbahnen oder elektrischen Autobahnen verwendbar usw., usw.

Wir können auch zwei Lichtschranken verwenden, indem das vorhandene Programm geringfügig geändert und ein zweiter Zähler programmiert wird.

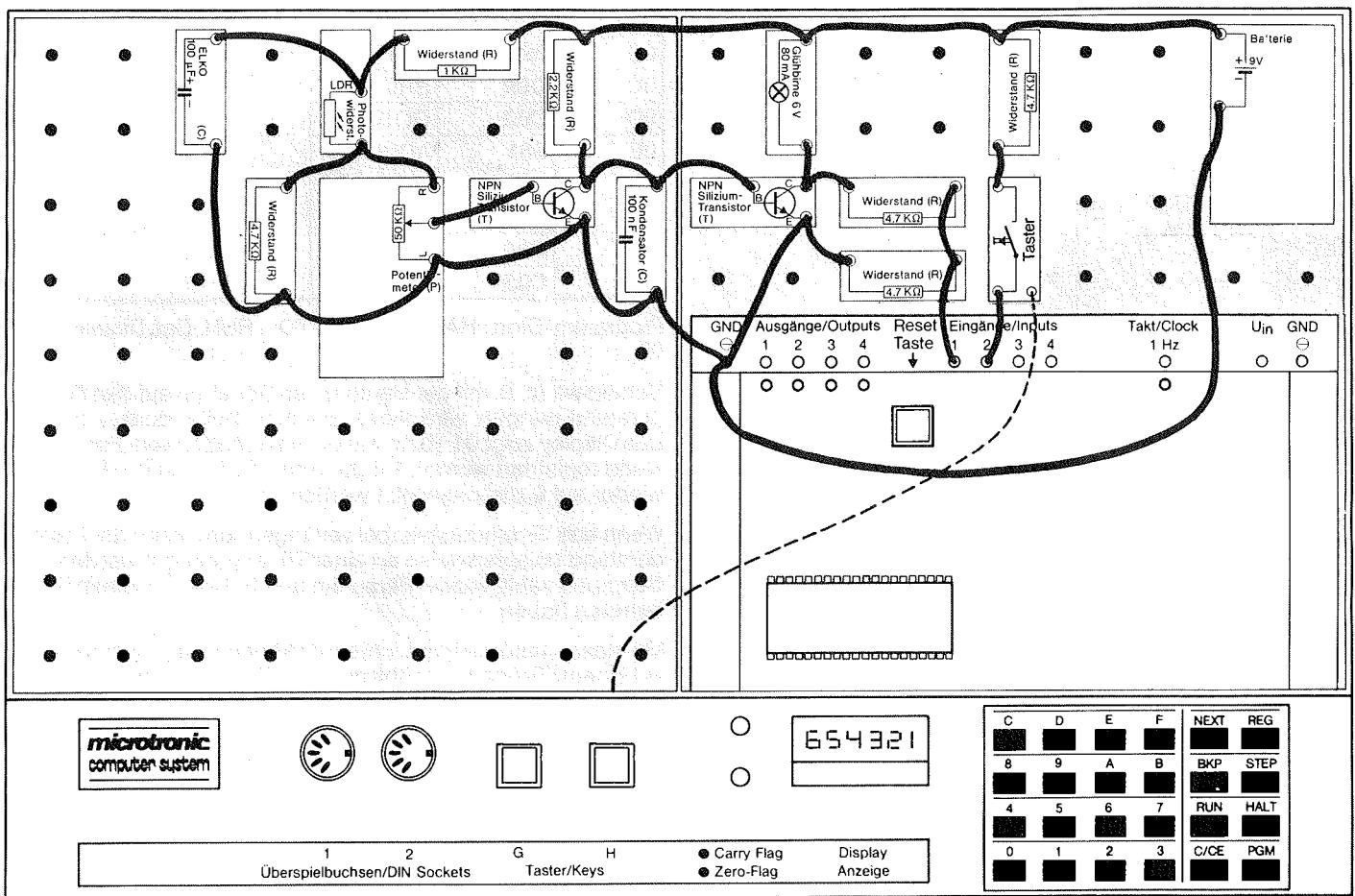
Werden zwei Lichtschranken nebeneinander, z. B. am Eingang eines Gebäudes montiert, könnte am Abend festgestellt werden, wieviel Personen das Gebäude betreten und wieder verlassen haben. Ein Vergleichs-Befehl könnte kontrollieren, auf welcher der beiden Lichtschranken zuerst ein Schatten registriert wurde, wobei der erste Schatten einen Aufwärts-Zähl-Effekt und der zweite Schatten einen Abwärts-Zähl-Effekt auslöst. Ergibt sich auf dem Display eine Differenz, wäre z. B. feststellbar, daß sich noch eine gewisse Anzahl von Personen im Gebäude befinden.

Auf Grund der bis jetzt erworbenen Kenntnisse sollte es mit einigen Überlegungen möglich sein, entsprechende Einsatzmöglichkeiten für Lichtschranken selbst zu programmieren.

Bei einer Autorennbahn ergibt sich die Möglichkeit, durch Einbeziehung des BUSCH-Relais 5964 bei einer bestimmten vorgegebenen Rundenzahl den Fahrstrom für die Autos selbsttätig abzuschalten, usw.



Schaltungsaufbau Lichtschranke



## Prinzip der Daten-Übertragung

(BUSCH-Digital-Studio 2075 erforderlich)

### Das interessiert den Programmierer:

Der Schaltungsaufbau der Lichtschranke ist uns aus dem Anleitungsbuch 2065, bzw. 2070 bekannt. Zusätzlich sind lediglich die beiden 4,7 KΩ Widerstände eingefügt, um einen möglichst geringen Stromverbrauch der Batterie zu erreichen.

Durch den Befehl DIN 0 wird der Eingangswert in Register 0 gespeichert. Durch ANDI 3,0 werden die dritte und vierte Dual-Stelle (also am Eingang 3 und 4) auf 0 gesetzt, damit z. B. ein am Eingang 4 angeschlossener Takt (falls wir nebenher das Uhren-Programm in Betrieb haben) die nachfolgenden Vergleiche nicht zerstört.

Anschließend erfolgt ein Vergleich, ob in den Registern 0 und 3 der gleiche Wert vorhanden ist. Wenn „ja“ hat sich der Eingangswert nicht geändert und es erfolgt ein Rücksprung zur Adresse 02. Wenn „nein“ wird der Eingangswert in den Registern 0 und 3 gespeichert. Durch den Vergleichs-Befehl CMPI wird festgestellt: Der Wert 1 steht in Register 0, d. h. die Lichtschranke wurde nicht unterbrochen und es erfolgt ein Sprung zur Adresse 02. Ist der Register-Inhalt größer als 1 wurde die rote Taste H betätigt und es erfolgt ein Sprung zur Adresse 00, um alle Register zu löschen. Andernfalls ergibt sich in Register 0 der Wert 0, d. h. die Lichtschranke wurde unterbrochen. Im folgenden dezimalen Zähler-Programm wird die Unterbrechung als Wert 1 dazu gezählt.

Warum ergibt sich bei unterbrochener Lichtschranke in Register 0 der Wert 0?

Die elektronische Schaltung ist so ausgelegt, daß bei nicht unterbrochener Lichtschranke am Eingang 1 des Computers eine kleine Spannung vorhanden ist – der Eingang ist „high“. Bei unterbrochener Lichtschranke (wenn das Lämpchen auf der Schaltung leuchtet) ist am Computer-Eingang keine Spannung vorhanden. Es ergibt sich ein „low“, also der Wert 0, welcher in Register 0 gespeichert wird.

Für die vielen interessanten Experimente mit elektronischen Peripherie-Schaltungen sind entsprechende Grundkenntnisse Voraussetzung. BUSCH-Electronic-Studios erklären eine breite Palette elektronischer Möglichkeiten, welche in Verbindung mit dem Computer zu immer neuen Einsatzmöglichkeiten führen.

In der Praxis ist es oft notwendig, Daten von einer EDV zu einer anderen zu übermitteln.

Mit dem IC-Zähler-Baustein aus dem Digital-Studio 2075 können wir einen Register-Inhalt unseres Computers auf der großen 7-Segment-Anzeige des Zähler-Bausteins sichtbar machen. Es ergeben sich zwei Möglichkeiten der Daten-Übertragung:

## Daten-Übertragung mit nur drei Leitungen

Wir geben wie üblich das Programm ein:

Adresse	Eingabe-	Mnemonic
00	<b>F08</b>	CLEAR
01	<b>132</b>	MOVI 3,2
02	<b>123</b>	MOVI 2,3
03	<b>114</b>	MOVI 1,4
04	<b>F10</b>	DISP 1,0
05	<b>FF0</b>	KIN 0
06	<b>101</b>	MOVI 0,1
07	<b>FE4</b>	DOT 4
08	<b>FE2</b>	DOT 2
09	<b>801</b>	CMP 0,1
0A	<b>E04</b>	BRZ 04
0B	<b>511</b>	ADDI 1,1
0C	<b>FE3</b>	DOT 3
0D	<b>C08</b>	GOTO 08

Der IC-Zähler-Baustein wird gemäß Abbildung am Computer angeschlossen.

Nach dem Programm-Start zeigt das Display 0. Wird eine Zahl eingegeben, wird diese sowohl auf dem Computer-Display als auch auf dem IC-Zähler-Baustein angezeigt.

Solange wir nur Zahlen eingeben, sind diese Zahlen auch auf der großen 7-Segment-Anzeige sichtbar. Geben wir beispielsweise C ein, erscheint auf der großen 7-Segment-Anzeige die Zahl 2.

Der IC-Zähler-Baustein arbeitet nur im dezimalen Bereich. Das hexadezimale C entspricht der dezimalen Zahl 12. Nachdem der IC-Zähler-Baustein nur einstellige Ergebnisse anzeigt, wird nur die letzte Stelle, nämlich die Ziffer 2 dargestellt.

Hätten wir mehrere IC-Zähler-Bausteine, wäre es ohne weiteres möglich, auch mehrstellige Ergebnisse zu übermitteln. Wie mehrere IC-Zähler-Bausteine zu schalten sind, ist aus dem Anleitungsbuch 2075 ersichtlich.

#### **Das interessiert den Programmierer:**

Aus dem Anleitungsbuch 2075 ist uns bekannt, daß der IC-Zähler-Baustein „nur“ zählen kann. Der Zähler-Baustein zählt immer um eins weiter, sobald an seinem Takt-Eingang die Spannung von „high“ nach „low“ wechselt. Der Zähler-Baustein kann auf Null zurückgesetzt werden, sobald sein RESET-Eingang „low“ wird.

Damit der IC-Zähler-Baustein die gleiche Zahl anzeigt, die auch im Computer eingegeben wird, werden hintereinander eine entsprechende Anzahl „high“ – „low“ – „high“ – Impulse an den IC-Zähler-Baustein weitergegeben, welche von diesem

mitgezählt werden, bis das Ergebnis angezeigt wird. Die Anzahl der Impulse wird vom Computer intern in Register Nr. 1 mitgezählt, wodurch der Computer weiß, welchen Zählerstand der IC-Zähler-Baustein hat.

Durch das eingegebene Programm wird zunächst in Register 2 der Wert 3 (dual: 0011) gespeichert. In Register 3 wird der Wert 2 (dual: 0010) und in Register 4 der Wert 1 (dual: 0001) gespeichert. Wird nun über die Tastatur in Register 0 ein Zahlenwert eingegeben, wird das Register 1 auf den Wert 0 gebracht und durch den Befehl DOT 4 (0001) und DOT 2 (0011) wird auch der IC-Zähler-Baustein auf 0 gesetzt.

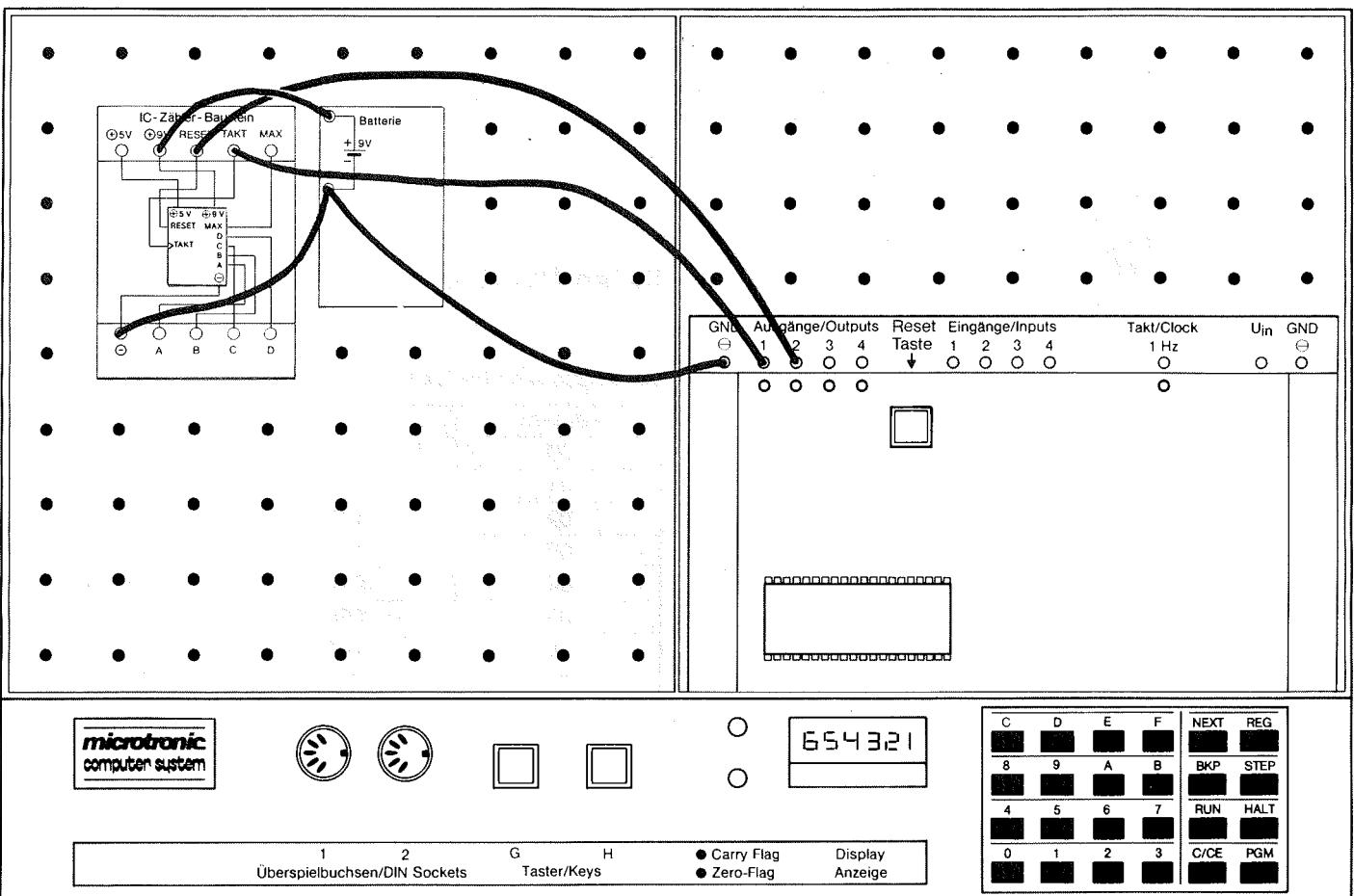
Nun folgt der Vergleich, ob der Inhalt von Register 0 mit dem Inhalt von Register 1 identisch ist. Wenn „ja“, dann ist der vom Zähler-Baustein angezeigte Wert identisch mit dem eingegebenen Wert. Wenn „nein“, wird zum Register 1 der Wert 1 addiert und durch den Befehl DOT 3 (Inhalt von Register 3 = dual 0010) der Ausgang 1 auf Null gesetzt. Durch die Änderung des „high“ – „low“ – Zustandes zählt der Zähler-Baustein um eins weiter und durch den Befehl DOT 2 (0011) wird auch der Ausgang 1 wieder „high“. Dieses Spiel wird solange fortgesetzt, bis die Inhalte der Register 0 und 1 gleich sind.

Dieses Verfahren hat zwei Nachteile.

Wird z. B. durch eine Störung die Leitungsverbindung zum IC-Baustein unterbrochen, ergibt sich dort eine falsche Ergebnisanzeige.

Oder wird beispielsweise am Computer der Wert 9 eingegeben, beginnt der IC-Zähler-Baustein von 0 an bis zum eingegebenen Wert hochzuzählen, wodurch ein gewisser Zeitbedarf notwendig ist. Der Vorteil des Verfahrens liegt jedoch darin, daß für die Datenübertragung lediglich drei Leitungen notwendig sind.

#### **Datenübertragung mit 3 Leitungen**



## Datenübertragung im „hand-shaking“-Verfahren

Zur Demonstration dieses Verfahrens dient das folgende Programm, welches wie üblich eingegeben wird:

Adresse	Eingabe-Befehl	Mnemonic
00	<b>F08</b>	CLEAR
01	<b>132</b>	MOVI 3,2
02	<b>123</b>	MOVI 2,3
03	<b>F10</b>	DISP 1,0
04	<b>FF0</b>	KIN 0
05	<b>FD1</b>	DIN 1
06	<b>801</b>	CMP 0,1
07	<b>E04</b>	BRZ 04
08	<b>FE3</b>	DOT 3
09	<b>FE2</b>	DOT 2
0A	<b>C05</b>	GOTO 05

Der IC-Zähler-Baustein wird gemäß Abbildung am Computer angeschlossen. Wir benötigen nun die doppelte Anzahl von Übertragungsleitungen gegenüber dem vorangegangenen Versuch.

Programm starten und eine Zahl eingegeben. Diese wird gegenüber dem vorangegangenen Experiment wesentlich schneller auf dem IC-Zähler-Baustein angezeigt. Geben wir nun einen Buchstaben ein, wird der Zähler-Baustein „ratlos“ – er zählt unendlich weiter, weil er keine Buchstaben darstellen kann.

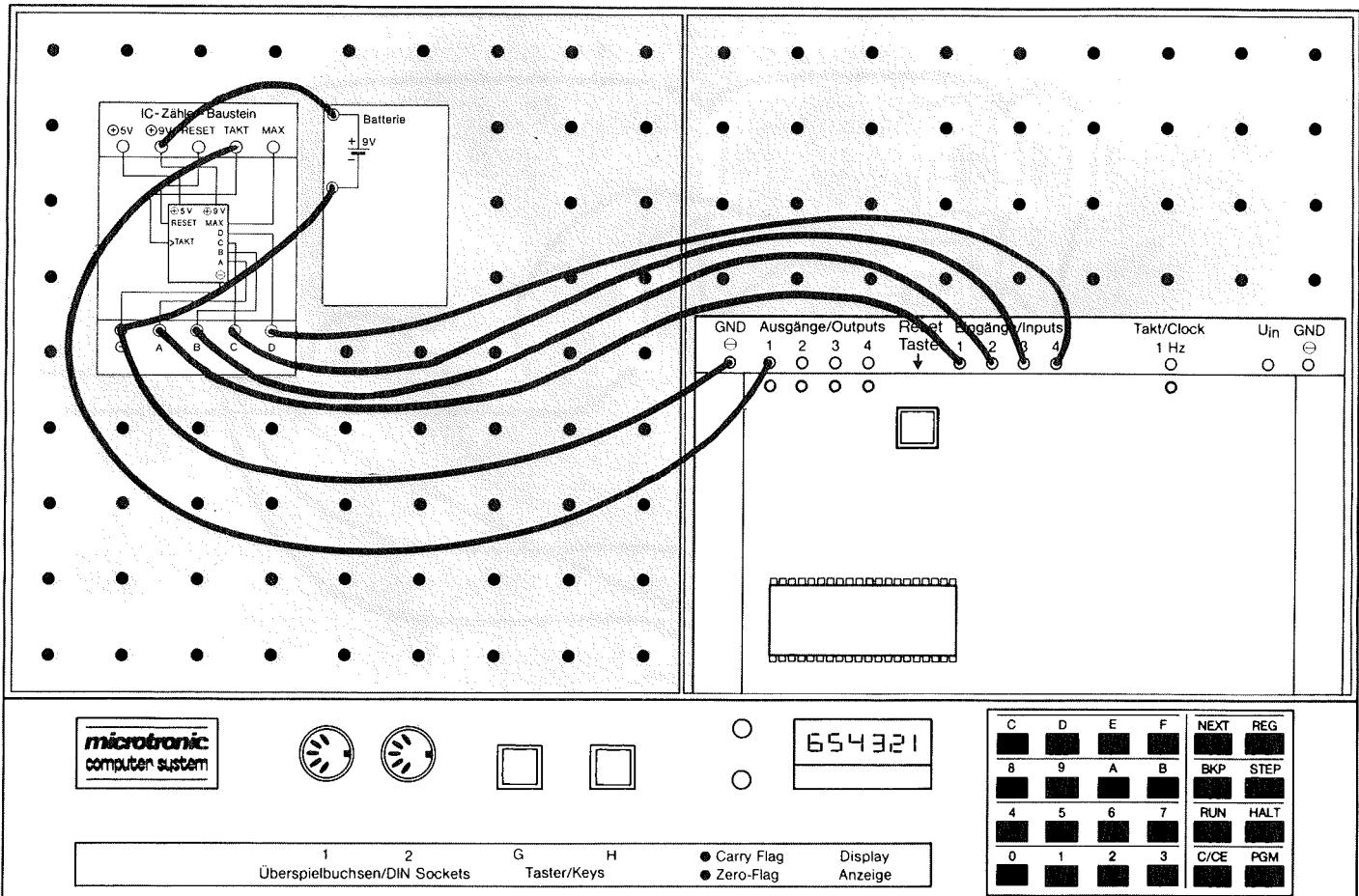
### Das interessiert den Programmierer:

Wir erkennen, daß vom Computer-Ausgang Nr. 1 eine Leitung zum Takt-Eingang des Zähler-Bausteins führt. Die vier Ausgänge des Zähler-Bausteins (A, B, C, D) sind mit den vier Computer-Eingängen verbunden. Die vier Ausgänge des Zähler-Bausteins geben die angezeigte Zahl als duale Werte weiter. Diese werden durch den Befehl DIN 1 in das Register 1 des Computers übernommen. Das eingegebene Programm vergleicht diese Zahl mit der über die Tastatur eingegebenen Zahl (im Register 0) und falls beide gleich sind, wurde das Ergebnis erreicht. Andernfalls wird durch die Befehle DOT 3 und DOT 2 der IC-Zähler um eins erhöht und der Vergleich wird wiederholt bis das Ergebnis erreicht wird. Somit ergibt sich vom IC-Zähler eine Rückmeldung zum Computer, wodurch eine falsche Datenübertragung ausgeschlossen wird.

Die Experimente zeigen, daß z. B. IC's direkt mit den Computer Ein- und Ausgängen verbunden werden können. Da die IC's mit geringen Spannungen arbeiten, sind keine zusätzlichen Verstärker-Schaltungen für die Ansteuerung erforderlich.

Übrigens, auch unser IC-Zähler-Baustein besitzt ein Übertrag-Signal (eine Art Carry-Flag) zur Ansteuerung eines zweiten Zähler-Bausteins. Der Ausgang MAX ist ständig „high“ und wird erst bei Erreichen des Wertes 9 „low“, um beim Übergang nach 0 wieder „high“ zu werden. Mit diesem Übertrag-Signal kann ein zweiter oder mehrere Zähler-Bausteine angesteuert werden.

## Datenübertragung im „hand-shaking“



# Computer zählt Frequenzen

(BUSCH-Electronic-Studio 2065 oder 2070 erforderlich).

Frequenz-Meßgeräte sind in der Elektronik zur Überprüfung von schwingungserzeugenden Schaltungen notwendig. Mit Frequenz-Meßgeräten lässt sich z. B. die Sendefrequenz eines Rundfunksenders oder auch die Tonfrequenz einer astabilen Kippstufe bestimmen.

Das Meßprinzip ist bei allen Frequenz-Zählern gleich. Während einer bestimmten Zeit (z. B. innerhalb einer Sekunde) wird die Anzahl der Frequenz-Impulse gezählt und ausgewertet. Auch unser Experiment arbeitet nach diesem Prinzip.

## Programm: Frequenz-Zähler

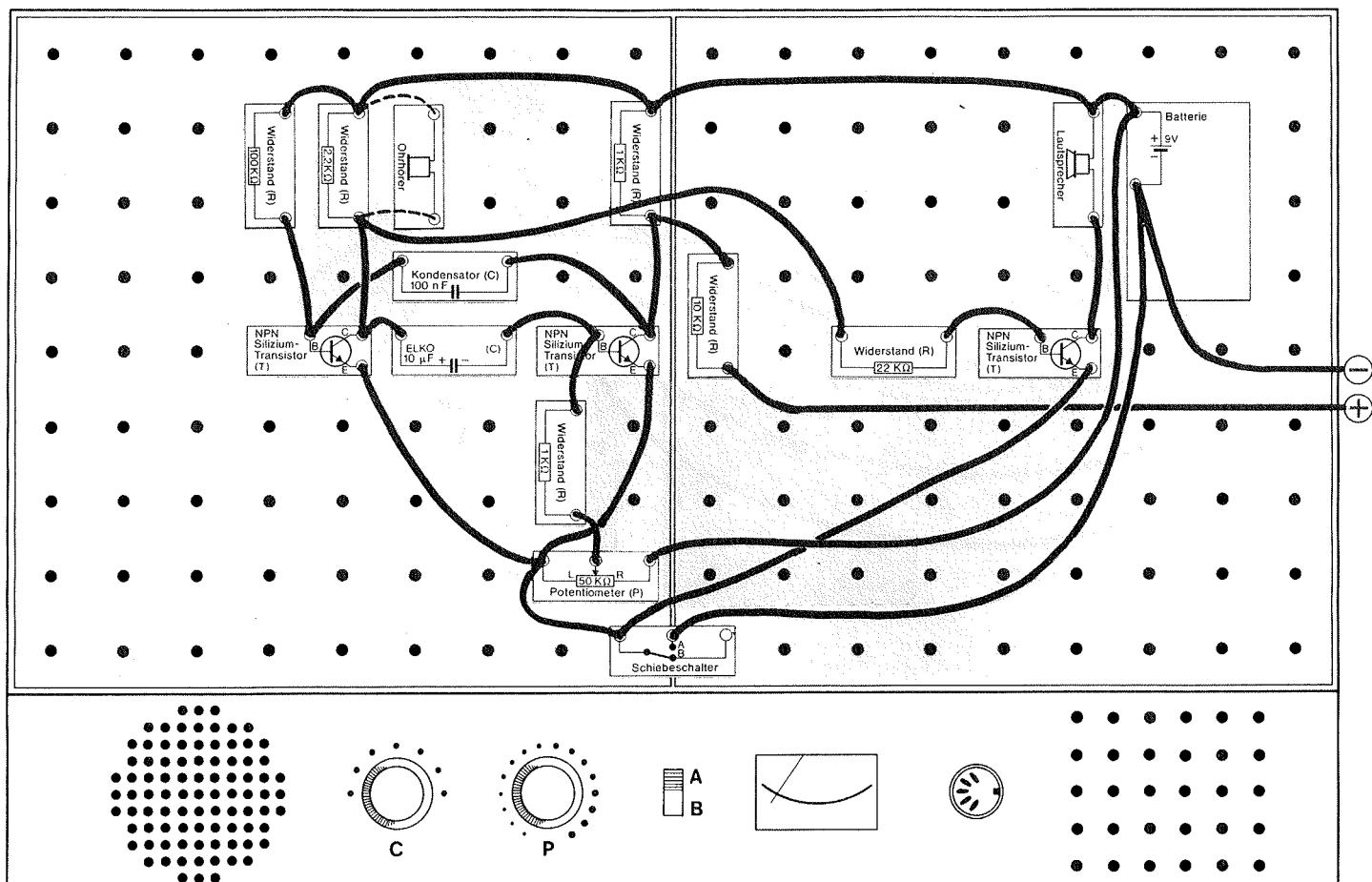
00	<b>F08</b>	CLEAR
01	<b>F28</b>	DISP 2,8
02	<b>FDF</b>	DIN F
03	<b>FAF</b>	SHL F
04	<b>E02</b>	BRZ 02
05	<b>FDF</b>	DIN F
06	<b>FAF</b>	SHL F
07	<b>E09</b>	BRZ 09
08	<b>C05</b>	GOTO 05
09	<b>F06</b>	TIME
0A	<b>F0F</b>	EXRA
0B	<b>FDF</b>	DIN F
0C	<b>FAF</b>	SHL F

0D	<b>E0B</b>	BRZ 0B
0E	<b>FDF</b>	DIN F
0F	<b>FAF</b>	SHL F
10	<b>E12</b>	BRZ 12
11	<b>C0E</b>	GOTO 0E
12	<b>F06</b>	TIME
13	<b>62A</b>	SUB 2,A
14	<b>D16</b>	BRC 16
15	<b>C18</b>	GOTO 18
16	<b>76A</b>	SUBI 6,A
17	<b>513</b>	ADDI 1,3
18	<b>63B</b>	SUB 3,B
19	<b>D1B</b>	BRC 1B
1A	<b>C1C</b>	GOTO 1C
1B	<b>7AB</b>	SUBI A,B

Bevor wir das Programm starten, ist die Schaltung gemäß Abbildung aufzubauen. Die mit  $\oplus$  bezeichnete Verbindungsleitung wird am EINGANG 4 des Computers angeschlossen. Die mit  $\ominus$  bezeichnete Leitung führt zur GND-Buchse des Computers.

An der Computer-Platine ist TAKT/CLOCK mit dem EINGANG 1 zu verbinden.

## Schaltung astabile Kippstufe (Multivibrator) für Frequenzmessung



Der Aufbauplan (Abbildung) ist maßgebend, wenn das große Electronic-Studio 2070 zur Verfügung steht. Ist lediglich das Electronic-Studio 2065 vorhanden, entfallen die 3 Bausteine, welche direkt links neben der Batterie dargestellt sind (Lautsprecher, Transistor und 22 KΩ Widerstand). In diesem Fall ist der Ohrhörer zu verwenden (im Plan mit punktierter Leitung eingezeichnet). Die Verbindungsleitung zwischen den beiden Widerständen 22 KΩ entfällt und die Verbindungsleitung vom Plus-Pol der Batterie führt direkt zum 1 KΩ Widerstand (Mitte oben).

Programm-Start und die elektronische Schaltung (Schiebeschalter Stellung A) in Betrieb nehmen. Solange der Potentiometerknopf ganz am linken Anschlag steht, ergibt sich im Lautsprecher (bzw. Ohr-Hörer) keine Tonerzeugung. Das Computer-Display zeigt 00.

Jetzt drehen wir den Potentiometerknopf etwas nach rechts bis im Lautsprecher (Ohr-Hörer) ein langsamen Knattern hörbar wird. Das Computer-Display zeigt die Frequenz des knatternden Tones in Hertz (Schwingungen pro Sekunde). Wird das Potentiometer weiter nach rechts gedreht, wird die Tonfolge schneller – die Frequenz nimmt zu, was auch am Display durch eine höhere Hertz-Zahl angezeigt wird.

Das Potentiometer sollte immer nur ein kleines Stück weitergedreht werden, weil der Computer bis zur Anzeige des Ergebnisses eine Meßzeit von ca. 2 Sekunden benötigt.

Je weiter der Potentiometerknopf nach rechts gedreht wird, umso schneller wird die Tonfolge, umso höher die Frequenzanzeige. Wir werden einen Punkt erreichen, an welchem der Computer nicht mehr mitzählt (oder schwankende Ergebnisse anzeigt), obwohl durch die schneller werdenden Schwingungen die Tonhöhe ansteigt und somit die Frequenzanzeige höher sein müßte.

Die von unserer Schaltung erzeugte Frequenz ist jetzt zu groß geworden (dies wird bei der Display-Anzeige zwischen 30 und 50 sein).

Für einen folgenden Versuch sollten wir uns merken, bis zu welcher Frequenzanzeige das Experiment funktioniert. Von einer möglichst genauen Angabe hängt die Genauigkeit des später folgenden Experiments ab.

Die Grenz-Frequenz wird ermittelt, indem beim kritischen Punkt das Potentiometer nur noch ein sehr kleines Stück weitergedreht wird, wobei die Anzeige zu beachten ist. Bei einer bestimmten Einstellung wird der angezeigte Wert stark schwanken. Die Potentiometer-Einstellung wird nun gerade so weit zurückgenommen, bis sich der angezeigte Wert nicht mehr als 3 Hz ändert. Diesen Wert halten wir uns in der folgenden Tabelle fest.

Die Grenz-Frequenz wurde mit: .... Hz ermittelt.

### Das interessiert den Programmierer:

Es ist nicht möglich, die Frequenzen durch ein einfaches Zähl-Programm zu bestimmen, indem die Impulse an einem Eingang gezählt werden. Unser Computer arbeitet für hohe Frequenzen zu langsam.

Bereits bei 3 bis 4 Impulsen (Schwingungen) pro Sekunde, kann der Computer nicht mehr mitzählen. Dies liegt daran, weil unser Mikroprozessor für sehr vielseitige Anwendungen programmiert ist. Er muß nicht nur zählen, er kontrolliert die Tastatur, er besorgt das im Background weiterlaufende Uhren-Programm und, und, und...

Durch einen einfachen Trick können wir den Computer überlisten: Im Betriebs-System läuft ständig die Uhrzeit mit. Auch dieses Uhren-Programm ist ein „Zähler“, welcher die Takt-Impulse am EINGANG 4 zählt und für die Uhrzeit auswertet.

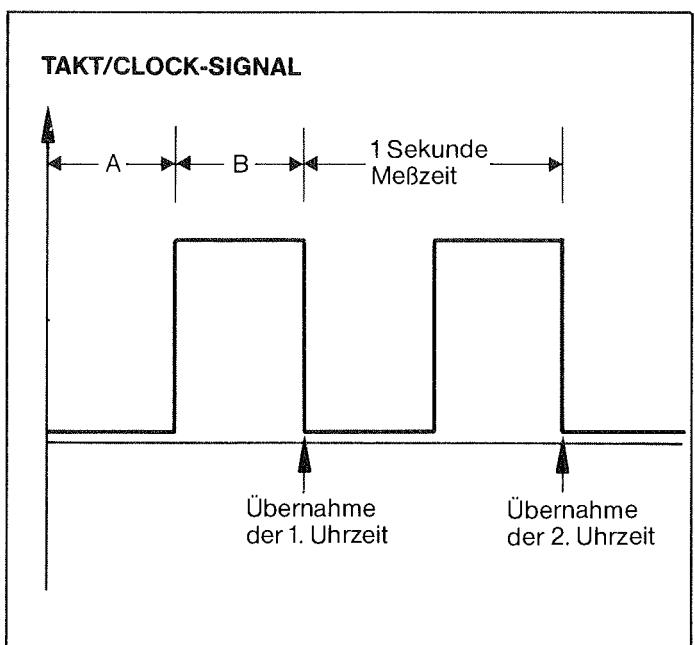
Dieser „Uhren-Zähler“ im Betriebs-System arbeitet verhältnismäßig schnell. Deshalb werden die zu messenden Impulse am EINGANG 4 angeschlossen. Dort werden die Impulse mitgezählt und zu einer Art „Uhrzeit“ weiterverarbeitet.

Dieser Vorgang kann leicht kontrolliert werden, indem wir mit HALT – PGM – 4 das Uhren-Programm aufrufen und gleichzeitig mit der angeschlossenen Schaltung einen Ton erzeugen. Wir können feststellen, daß je nach Takt-Frequenz der Schaltung die Sekundenanzeige auf dem Display mit erheblicher Geschwindigkeit angezeigt wird.

Das Computer-Programm bringt die „Uhrzeit“ durch den TIME-Befehl in die Register A bis F, wobei die Werte durch den EXRA-Befehl in Register 2 bis 7 gespeichert werden. Genau eine Sekunde später wird erneut die „Uhrzeit“ übernommen und gespeichert. Subtrahiert man die beiden „Uhrzeiten“ voneinander, ergibt sich die Frequenz in Hertz (Schwingungen pro Sekunde).

Es muß noch sichergestellt werden, daß die Meßzeit zwischen den beiden TIME-Befehlen genau eine Sekunde beträgt. Hierfür werden die Impulse des 1 Hz-TAKT/CLOCK-Ausgangs verwendet. Sobald das Programm gestartet wird, wartet der Computer in einer Warteschleife, bis das Takt-Signal „high“ (1) vorhanden ist (Siehe Abbildung, Abschnitt A).

In einer zweiten Warteschleife wartet der Computer bis das Takt-Signal wieder „low“, also 0 wird (in der Abbildung Abschnitt B).



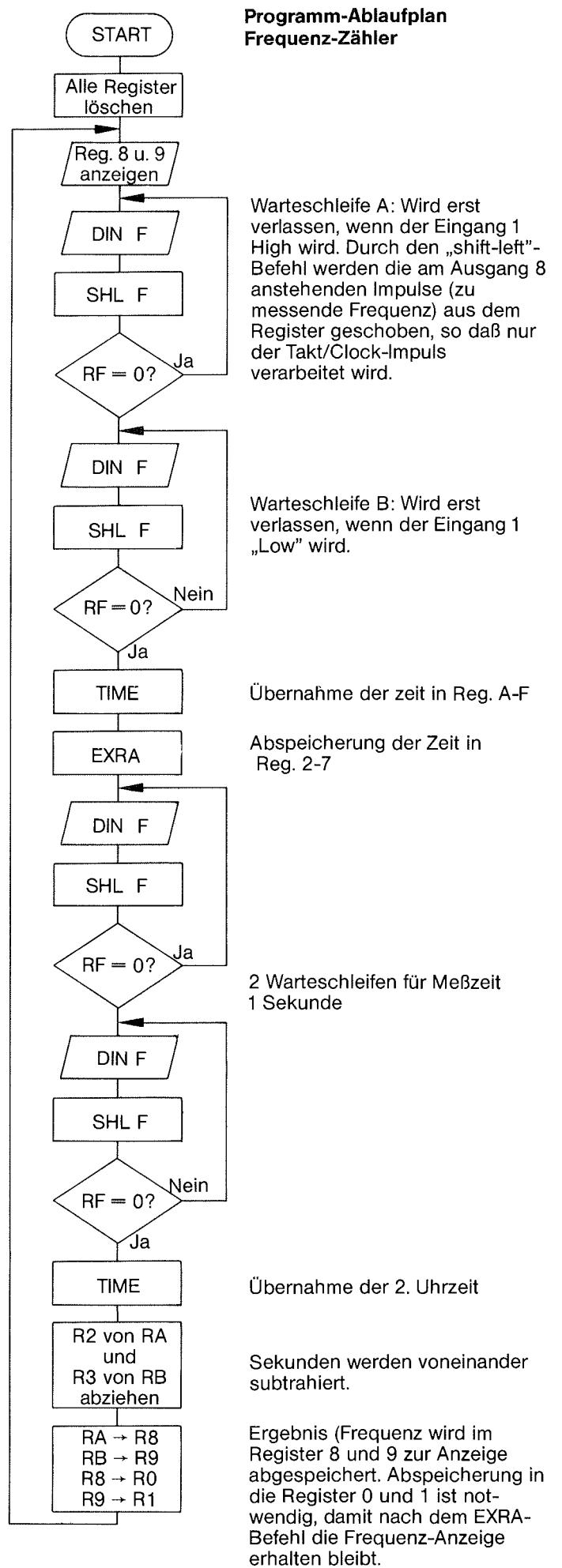
## Programm-Ablaufplan Frequenz-Zähler

Im gleichen Augenblick, wenn das Takt-Signal „low“ (0) wird, übernimmt der Computer die erste Uhrzeit. Mit der zweiten Warteschleife wird erreicht, daß der Signal-Wechsel nach „high“ und wieder zurück zu „low“ abgewartet wird (genau 1 Sekunde), um danach die zweite Uhrzeit zu übernehmen und auszuwerten.

Die beiden Warteschleifen (Abschnitt A und B in der Abbildung) sind erforderlich, um einen definierten Anfang der Meßzeit zu erreichen. Die einzelnen Programm-Funktionen ergeben sich aus dem Programm-Ablaufplan. Es werden nicht die vollständigen „Uhrzeiten“ voneinander abgezogen, sondern lediglich die Sekunden. Bei der Subtraktion der 10-er-Sekundenstelle wird der Dezimalausgleich beim Auftreten eines Übertrages durch die Subtraktion des hexadezimalen Wertes A (dezimal 10) durchgeführt, weil die 10-er Sekunden nur bis 5 gehen.

Der elektronische Aufbau des Multivibrators (auch astabile Kippstufe genannt) sollte aus den Experimenten mit den Electronic-Studios 2065, bzw. 2070 bekannt sein. Bei unserem Frequenzmeß-Experiment wird der Lautsprecher mit einem zusätzlichen Transistor betrieben. Dies ist erforderlich, weil der Computer für den Zählvorgang einwandfreie Rechteck-Impulse benötigt. Würde man den Lautsprecher ohne den zweiten Transistor betreiben, könnten die Spulenwicklungen des Lautsprechers die Rechteck-Impulse stark verfälschen.

Die Anordnung der elektronischen Multivibrator-Schaltung bleibt auch für das folgende Experiment bestehen.



# Ein hochinteressantes Experiment

## Reaktionszeit-Meßgerät

Der Multivibrator (astabile Kippstufe) des vorangegangenen Experiments bleibt unverändert bestehen. An der Computer-Platine sind die Anschlüsse gemäß Abbildung herzustellen. Die Plus- und Minus-Leitungen des Multivibrators zur Computer-Platine sind ebenfalls aus der Abbildung ersichtlich. Der Piezo-Summer wird erst nach der Programmierung angeschlossen. Das Programm wird wie üblich eingegeben.

### Programm: Reaktionszeit-Meßgerät

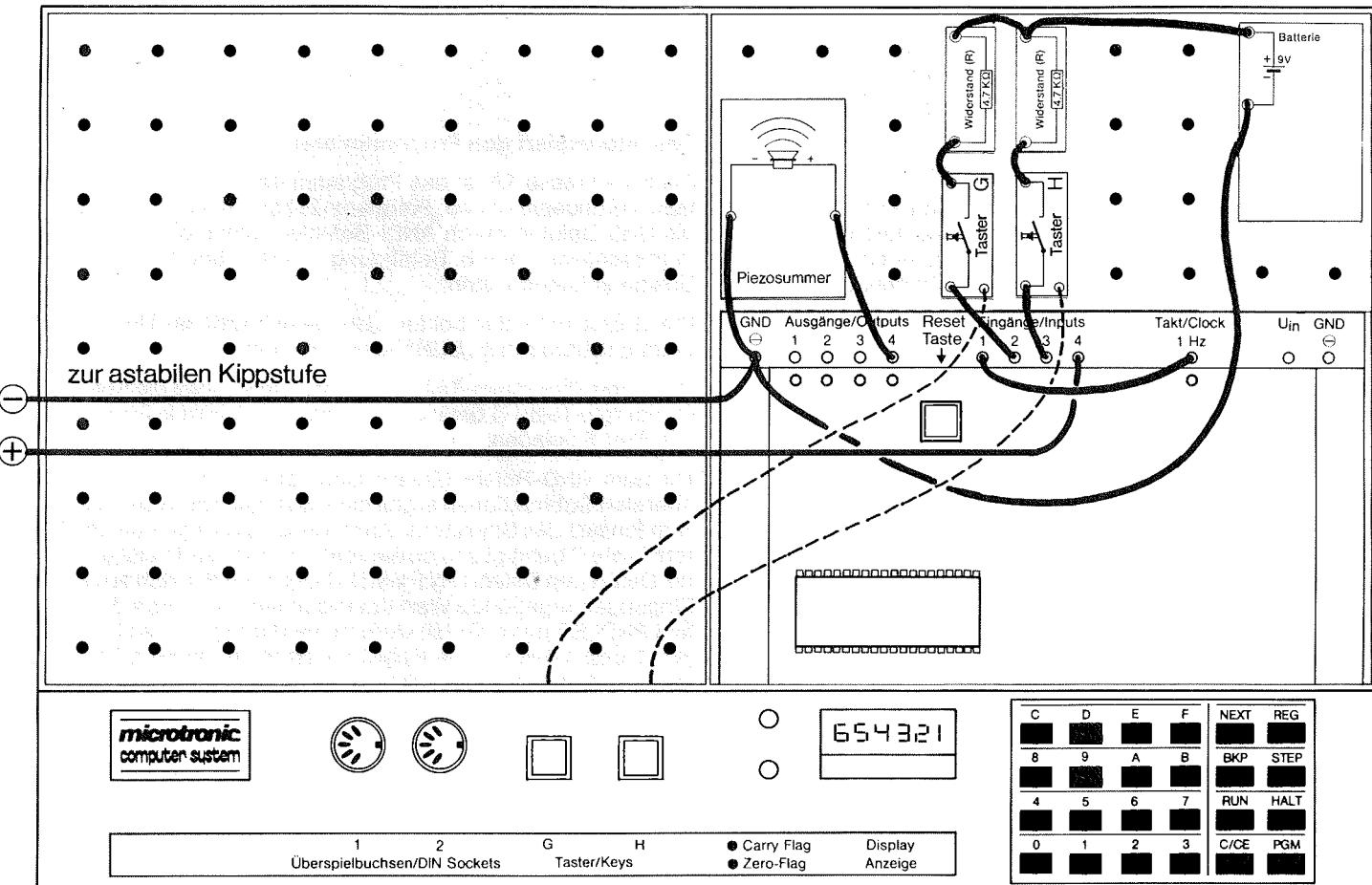
Adresse	Eingabe-Befehl	Mnemonic
00	<b>F08</b>	CLEAR
01	<b>F0E</b>	EXRM
02	<b>F08</b>	CLEAR
03	<b>FE0</b>	DOT 0
04	<b>F28</b>	DISP 2,8
05	<b>FDF</b>	DIN F
06	<b>31F</b>	ANDI 1,F
07	<b>E05</b>	BRZ 05
08	<b>FDF</b>	DIN F
09	<b>31F</b>	ANDI 1,F
0A	<b>E0C</b>	BRZ 0C
0B	<b>C08</b>	GOTO 08
0C	<b>F06</b>	TIME
0D	<b>F0F</b>	EXRA
0E	<b>FDF</b>	DIN F
0F	<b>31F</b>	ANDI 1,F
10	<b>E0E</b>	BRZ 0E

11	<b>FDF</b>	DIN F
12	<b>31F</b>	ANDI 1,F
13	<b>E15</b>	BRZ 15
14	<b>C11</b>	GOTO 11
15	<b>F06</b>	TIME
16	<b>B78</b>	CALL 78
17	<b>0A8</b>	MOV A,8
18	<b>0B9</b>	MOV B,9
19	<b>080</b>	MOV 8,0
1A	<b>091</b>	MOV 9,1
1B	<b>FDF</b>	DIN F
1C	<b>32F</b>	ANDI 2,F
1D	<b>E04</b>	BRZ 04
1E	<b>158</b>	MOVI 5,8
1F	<b>F02</b>	DISOUT
20	<b>F05</b>	RND
21	<b>71D</b>	SUBI 1,D
22	<b>FCE</b>	SUBC E

23	<b>D25</b>	BRC 25
24	<b>C21</b>	GOTO 21
25	<b>FDF</b>	DIN F
26	<b>36F</b>	ANDI 6,F
27	<b>92F</b>	CMPI 2,F
28	<b>E31</b>	BRZ 31
29	<b>1F0</b>	MOVI F,0
2A	<b>1F1</b>	MOVI F,1
2B	<b>FE1</b>	DOT 1
2C	<b>101</b>	MOVI 0,1
2D	<b>FE1</b>	DOT 1
2E	<b>710</b>	SUBI 1,0
2F	<b>E20</b>	BRZ 20
30	<b>C2A</b>	GOTO 2A
31	<b>1FF</b>	MOVI F,F
32	<b>FEF</b>	DOT F
33	<b>F06</b>	TIME
34	<b>F0F</b>	EXRA
35	<b>FDF</b>	DIN F
36	<b>34F</b>	ANDI 4,F
37	<b>E35</b>	BRZ 35
38	<b>F06</b>	TIME
39	<b>10F</b>	MOVI 0,F
3A	<b>FEF</b>	DOT F
3B	<b>B78</b>	CALL 78
3C	<b>008</b>	MOV 0,8

Programm-Fortsetzung  
nächste Seite beachten!

### Anschluß der astabilen Kippstufe (von Seite 62) an Computer



Adresse	Eingabe-Befehl	Mnemonic
56	D58	BRC 58
57	C5A	GOTO 5A
58	569	ADDI 6,9
59	51A	ADDI 1,A
5A	42A	ADD 2,A
5B	D5F	BRC 5F
5C	99A	CMPI 9,A
5D	D5F	BRC 5F
5E	C61	GOTO 61
5F	56A	ADDI 6,A
60	51B	ADDI 1,B
61	F0E	EXRM
62	F30	DISP 3,0
63	FDF	DIN F
64	32F	ANDI 2,F
65	E63	BRZ 63
66	718	SUBI 1,8
67	E69	BRZ 69
68	C1F	GOTO 1F
69	F02	DISDUT
6A	F0E	EXRM
6B	F0F	EXRA
6C	F0D	EXRL
6D	150	MDVI 5,0
6E	101	MOVI 0,1
6F	102	MOVI 0,2
70	103	MOVI 0,3
71	104	MOVI 0,4
72	F0C	DIV
73	1E3	MOVI E,3

74	1E4	MOVI E,4
75	F50	DISP 5,0
76	FF0	KIN 0
77	C00	GOTO 00
78	62A	SUB 2,A
79	D7B	BRC 7B
7A	C7D	GOTO 7D
7B	76A	SUBI 6,A
7C	513	ADDI 1,3
7D	63B	SUB 3,B
7E	D80	BRC 80
7F	C82	GOTO 82
80	7AB	SUBI A,B
81	514	ADDI 1,4
82	10F	MDVI 0,F
83	64C	SUB 4,C
84	D86	BRC 86
85	C87	GDTO 87
86	76C	SUBI 6,C
87	90C	CMPI 0,C
88	E91	BRZ 91
89	71C	SUBI 1,C
8A	56B	ADDI 6,B
8B	99B	CMPI 9,B
8C	D8E	BRC 8E
8D	C87	GOTO 87
8E	56B	ADDI 6,B
8F	51F	ADDI 1,F
90	C87	GDTO 87
91	0FC	MOV F,C
92	F07	RET

Piezo-Summer anschließen. Programm starten.

Der Computer arbeitet zunächst als Frequenz-Zähler. Am Potentiometerknopf wird die Frequenz so eingestellt, bis das Display genau 20 Hz anzeigt. Bei evtl. Schwankungen der Anzeige muß solange nachjustiert werden, bis die 20 genau angezeigt wird.

Mit diesem Programm sind wir in der Lage, unsere Reaktionszeit sehr genau zu messen. Vielleicht ist es interessant, auch einmal eine Messung unter ungünstigen Voraussetzungen (unausgeschlafen, Alkoholgenuß) zu wiederholen.

Wir betätigen den roten Taster G und halten ihn im niedergedrückten Zustand, bis sich der Computer mit dem zufällig einsetzenden Signalton meldet. Nun muß möglichst schnell die rote Taste H gedrückt werden. Das Display zeigt 3-stellig die Reaktionszeit an. Die linke Display-Stelle zeigt die Sekunden, die mittlere die 1/10 Sekunden und die rechte Display-Stelle die 1/100 Sekunden. „040“ bedeutet also 0,4 Sekunden.

Es sind nacheinander 5 Reaktionszeit-Messungen vorgesehen, weshalb erneut die rote Taste G bis zum Beginn des Signaltons gedrückt wird. Der Wechsel auf die rote Taste H bringt die zweite Messung.

Nach der fünften Messung zeigt das Display: EE XXX. Für „X“ werden 3 Zahlen angezeigt – ein Durchschnittswert der fünf Reaktionszeit-Messungen.

Wird anschließend eine beliebige Zahlentaste betätigt, erscheint auf dem Display wieder die Frequenz-Messung, und wir können kontrollieren, ob die eingestellte Frequenz-Zahl „20“ noch stimmt. Sobald wieder die rote Taste G gedrückt wird, beginnt die nächste Reaktionszeit-Messung.

**Wird vor Ertönen des Signaltons** die rote Taste G nicht betätigt, oder die Taste H vorzeitig betätigt, ertönt ein unterbrochener Signalton zur Demonstration, daß die Messung nicht korrekt ausgeführt wurde. Unkorrekte Messungen werden nicht gewertet und auch bei der Ermittlung der Durchschnittszeit nicht mitgerechnet.

Fehlanzeigen (falsche Zeiten) können auftreten, wenn die Reaktionszeit einer Versuchsperson mehrere Sekunden beträgt (was hoffentlich in unserem Falle nicht zutrifft).

Beim vorangegangenen Experiment „Frequenz-Meßgerät“ haben wir die Grenzfrequenz unseres Computers festgestellt. Diese kann bei 30 Hz (mitunter sogar bei 40 oder 50 Hz) liegen.

Momentan ist unser Reaktionszeit-Meßgerät auf 20 Hz justiert. Damit liegt die Genauigkeit bei einer 1/20 Sekunde. Bringt unser Computer eine Grenzfrequenz von 30 oder 40 oder vielleicht sogar 50 Hz ergibt sich eine noch genauere Reaktionszeit-Messung, weil wir dann mit 1/30, 1/40 oder 1/50 Sekunden operieren können.

Bei einem Grenzwert 30 Hz ist eine Programm-Änderung bei Adresse 44 erforderlich. Der vorhandene Befehl MOVI 2,0 (Eingabe 120) ist in MOVI 3,0 (Eingabe 130) zu ändern. Der Multivibrator ist so zu justieren, daß das Display genau „30“ zeigt.

Lag unsere Grenzfrequenz bei oder über 40 Hz wird der MOVI-Befehl auf die Eingabe 140 geändert. Der Multivibrator ist so zu justieren, daß auf dem Display genau „40“ angezeigt wird. Bei 50 Hz lautet die MOVI-Eingabe 150 und das Display muß „50“ zeigen.

#### Das interessiert den Programmierer:

Bis zur Adresse 1A ist das Programm ähnlich aufgebaut, wie beim vorangegangenen „Frequenz-Zähler“. Allerdings wurden die SHL-Befehle durch ANDI-Befehle ersetzt, damit bei den Warteschleifen durch Betätigung eines roten Tasters kein Unsinn entstehen kann.

Die Subtraktion der beiden „Uhrzeiten“ wird im Unter-Programm (Subroutine) „SUB“ vorgenommen.

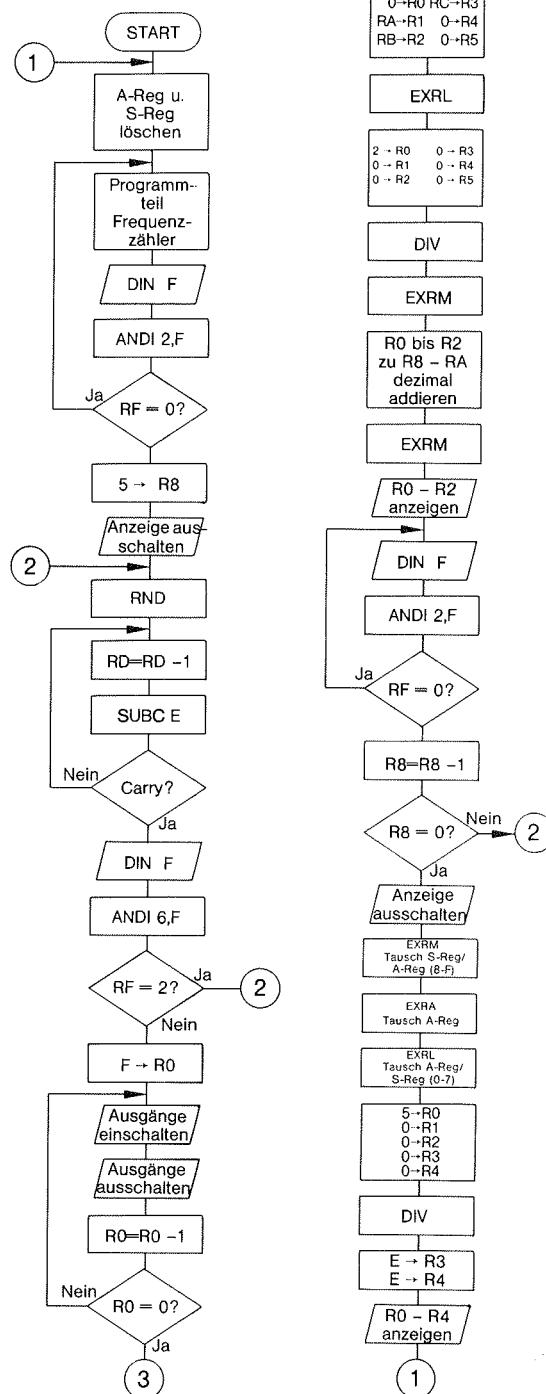
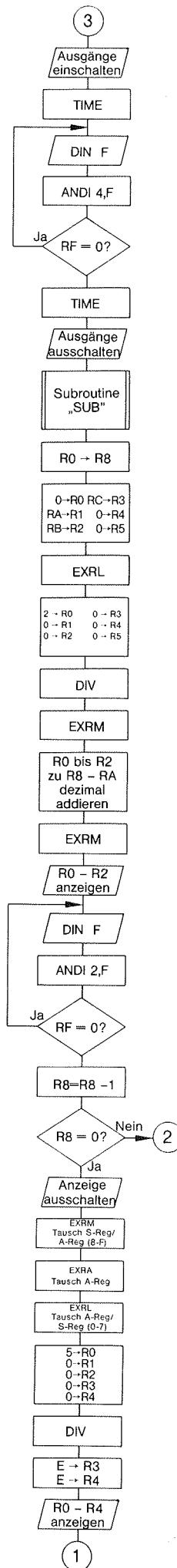
Nach dem Programm-Teil „Frequenz-Zähler“ folgt die Abfrage, ob die rote Taste G betätigt wird. Wenn „ja“, wird in Register 8 der Wert 5 geladen.

Mit dem RND-Befehl (Zufalls-Generator) und den entsprechenden Subtraktionen ergibt sich eine zufällige Wartezeit bis zum Einsatz des Signaltons. Anschließend wird geprüft, ob die rote Taste G betätigt und ob die Taste H noch nicht betätigt wurde. Durch den Befehl DIN F wird in Register F der sich an den 4 Eingängen ergebende Wert übernommen und durch den Befehl ANDI 6,F (dual: 0110) werden die Signale an den Eingängen 1 und 4 gelöscht. In Register F ist noch der Wert 2 (dual: 0010) vorhanden, wenn lediglich die Taste G betätigt wurde. Ist dies nicht der Fall, werden die Ausgänge mehrere Male ein- und ausgeschaltet (schnell unterbrochener Signalton als Demonstration, daß Tastenbedienung nicht korrekt vorgenommen wurde). Andernfalls wird der Dauerton durch den Befehl MOVI F,F und DOT F eingeschaltet und die erste „Uhrzeit“ wird übernommen. In einer Warteschleife wird die Betätigung des Tasters H abgewartet. Anschließend wird die zweite „Uhrzeit“ ebenfalls übernommen.

Im Unter-Programm „SUB“ wird durch Subtraktion der beiden Uhrzeiten die Anzahl der gezählten Impulse (Frequenz-Messung) festgestellt. Bei einer Frequenz-Einstellung von 20 Hz wird die gemessene Impulszahl durch 2 dividiert, wodurch die Reaktionszeit festgestellt wird. (bei einer Frequenz von 40 Hz wird durch 4 geteilt usw. – Befehls-Änderung bei Adresse 44).

Die ermittelte Reaktionszeit wird in den Speicher-Registern 8 bis A addiert, d. h. daß dort nach fünf Reaktionszeit-Messungen die Gesamtzeit gespeichert ist.

Das Programm wartet, bis die Taste G betätigt wird. Dann wird jeweils von Register 8 (in welchem beim Start der Wert 5 vorhanden war) der Wert 1 abgezogen. Die Messungen werden beendet, sobald im Register 8 der Wert 0 erreicht wird. Die Summe der Reaktionszeiten (in den Speicher-Registern 8 bis A) wird durch 5 dividiert und die durchschnittliche Reaktionszeit wird angezeigt.



# Digital-Voltmeter

(BUSCH-Electronic-Studios 2065 oder 2070 erforderlich)

Ein Digital-Voltmeter ist ein Spannungs-Meßgerät, bei welchem die Meßergebnisse digital, d. h. in direkt ablesbaren Zahlenwerten angezeigt werden. Gegenüber einem Analog-Meßgerät, bei welchem das Meßergebnis durch Zeigerausschlag angezeigt wird, sind digitale Anzeigen leichter ablesbar. Außerdem arbeiten digitale Meßgeräte genauer. Es ist allerdings auch ein größerer Schaltungsaufwand notwendig.

Bei einem Digital-Voltmeter muß die Spannung (eine analoge Größe) in eine digitale Größe umgewandelt werden. Hierfür gibt es spezielle integrierte Schaltkreise, sogenannte A/D-Wandler (Analog-Digital-Wandler).

Die Anzeigengenauigkeit eines Digital-Meßgerätes hängt davon ab, in wieviel Teilstücke die zu messende Größe unterteilt werden kann. Wird z. B. eine 10 V-Spannung in 20 Teilstücke unterteilt, ergibt sich als kleinste angezeigte Größe (oder als geringster Spannungsunterschied) ein Bereich von 0,5 V. Werden 10 V in 100 Teilstücke aufgeteilt, ergibt sich bereits eine Genauigkeit von 0,1 V.

Für die Analog-/Digital-Umwandlung verwenden wir wieder eine „astabile Kippstufe“, deren Aufbau sich jedoch gegenüber dem bisher verwendeten Multivibrator ändert. Wir bauen die Schaltung gemäß Abbildung neu auf. TAKT/CLOCK muß wieder mit EINGANG 1 verbunden werden.

Der Aufbauplan ist maßgebend, wenn das große Electronic-Studio 2070 zur Verfügung steht. Bei Verwendung des Electronic-Studios 2065 entfallen die 3 Bausteine, welche direkt links neben der Batterie dargestellt sind (Lautsprecher, Transistor und 22 K Ohm Widerstand). In diesem Fall ist der Ohrhörer zu verwenden, (im Plan mit punktierter Leitung eingezeichnet). Die Verbindungsleitungen zu den nicht benutzten Bausteinen

entfallen und der Plus-Pol der Batterie wird direkt am 1 K Ohm Widerstand angeschlossen (Mitte oben).

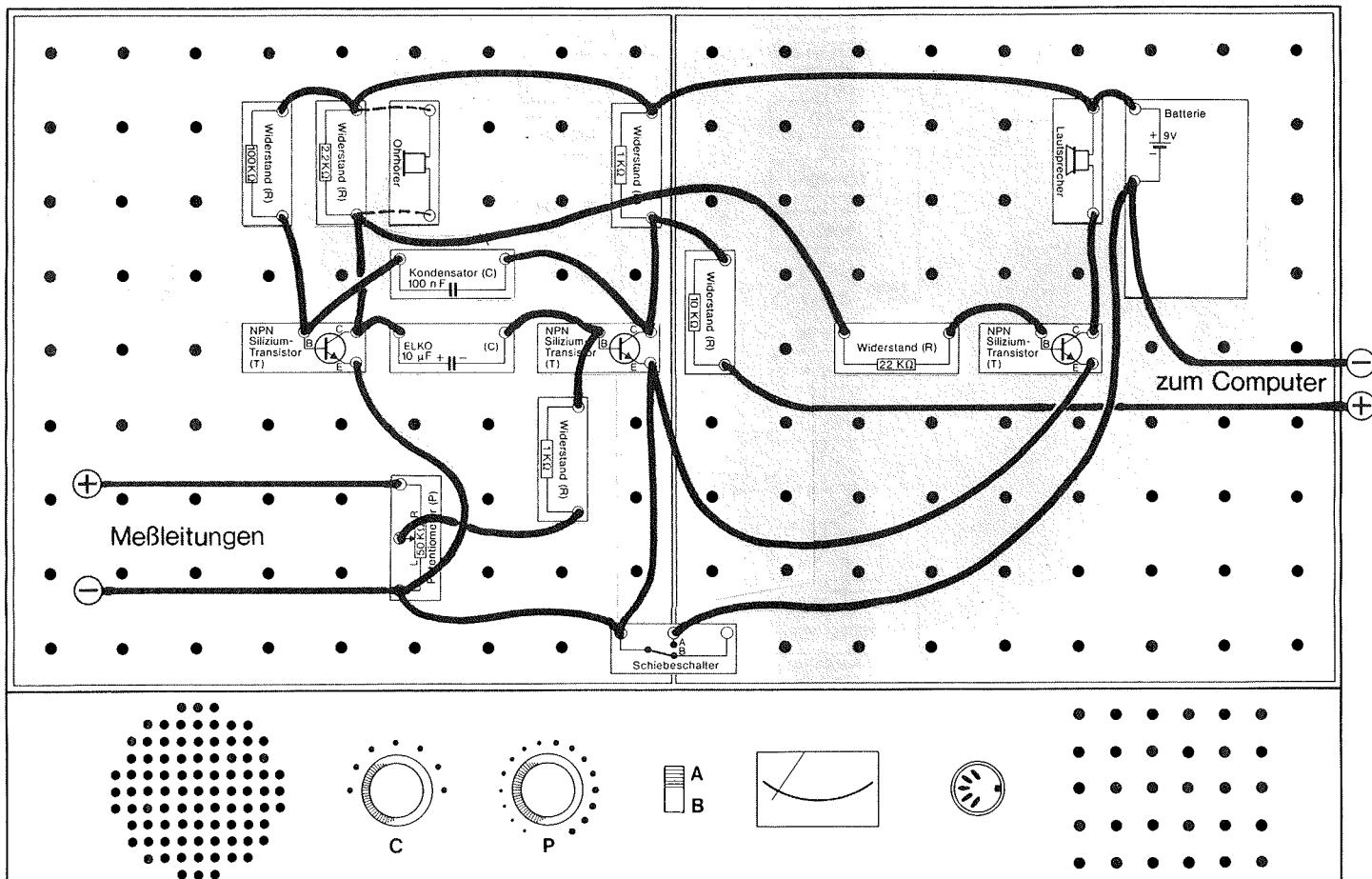
Die zum Computer führenden Leitungen sind wieder mit  $\oplus$  und  $\ominus$  gekennzeichnet. Wie beim vorangegangenen Versuch, wird die  $\oplus$  Leitung wieder am EINGANG 4 der Computer-Platine angeschlossen. Die  $\ominus$  Leitung führt an die GND-Buchse. Die evtl. noch angeschlossenen roten Tasten G und H sind nicht erforderlich.

Damit wir unser Digital-Voltmeter „eichen“ können, muß zunächst das Programm gemäß Tabelle eingegeben werden.

## Programm: Digital-Voltmeter

Adr.	Eingabe-Befehl	Mnemonic	Erklärungen
00	<b>F08</b>	CLEAR	
01	<b>F28</b>	DISP 2,8	
02	<b>FDF</b>	DIN F	
03	<b>FAF</b>	SHL F	
04	<b>E02</b>	BRZ 02	
05	<b>FDF</b>	DIN F	
06	<b>FAF</b>	SHL F	
07	<b>E09</b>	BRZ 09	
08	<b>C05</b>	GOTO 05	
09	<b>F06</b>	TIME	
0A	<b>F0F</b>	EXRA	
0B	<b>FDF</b>	DIN F	
0C	<b>FAF</b>	SHL F	
0D	<b>E0B</b>	BRZ 0B	

## Astabile Kippstufe für Digital-Voltmeter



0E	<b>FDF</b>	DIN F	
0F	<b>FAF</b>	SHL F	
10	<b>E12</b>	BRZ 12	
11	<b>C0E</b>	GOTO 0E	
12	<b>F06</b>	TIME	
13	<b>62A</b>	SUB 2,A	
14	<b>D16</b>	BRC 16	
15	<b>C18</b>	GOTO 18	
16	<b>76A</b>	SUBI 6,A	
17	<b>513</b>	ADDI 1,3	
18	<b>63B</b>	SUB 3,B	
19	<b>D1B</b>	BRC 1B	
1A	<b>C1C</b>	GOTO 1C	
1B	<b>7AB</b>	SUBI A,B	
1C	<b>B4C</b>	CALL 4C	Sprung nach „MOVI 0“
1D	<b>0A0</b>	MOV A,0	
1E	<b>0B1</b>	MOV B,1	
1F	<b>F0D</b>	EXRL	
20	<b>B4C</b>	CALL 4C	Sprung nach „MOVI 0“
21	<b>0A0</b>	MOV A,0	
22	<b>0B1</b>	MOV B,1	
23	<b>F0B</b>	MULT	
24	<b>F0D</b>	EXRL	
*25	<b>170</b>	MOVI 7,0	(MOVI 5,0 <b>150</b> )
*26	<b>161</b>	MOVI 6,1	(MOVI 2,1 <b>121</b> )
27	<b>F0B</b>	MULT	
28	<b>04F</b>	MOV 4,F	
29	<b>03E</b>	MOV 3,E	
2A	<b>02D</b>	MOV 2,D	
2B	<b>B4C</b>	CALL 4C	Sprung nach „MOVI 0“
2C	<b>0A0</b>	MOV A,0	
2D	<b>0B1</b>	MOV B,1	
2E	<b>F0D</b>	EXRL	
*2F	<b>161</b>	MOVI 6,1	(MOVI 5,1 <b>151</b> )
*30	<b>112</b>	MOVI 1,2	(MOVI 2,2 <b>122</b> )
*31	<b>133</b>	MOVI 3,3	(MOVI 1,3 <b>113</b> )
32	<b>F0B</b>	MULT	
33	<b>42D</b>	ADD 2,D	
34	<b>D38</b>	BRC 38	
35	<b>99D</b>	CMPI 9,D	
36	<b>D38</b>	BRC 38	
37	<b>C3A</b>	GOTO 3A	
38	<b>56D</b>	ADDI 6,D	
39	<b>51E</b>	ADDI 1,E	
3A	<b>43E</b>	ADD 3,E	
3B	<b>D3F</b>	BRC 3F	
3C	<b>99E</b>	CMPI 9,E	
3D	<b>D3F</b>	BRC 3F	
3E	<b>C41</b>	GOTO 41	
3F	<b>56E</b>	ADDI 6,E	
40	<b>51F</b>	ADDI 1,F	
41	<b>44F</b>	ADD 4,F	
42	<b>D46</b>	BRC 46	
43	<b>99F</b>	CMPI 9,F	

44	<b>D46</b>	BRC 46	
45	<b>C47</b>	GOTO 47	
46	<b>56F</b>	ADDI 6,F	
47	<b>0F9</b>	MOV F,9	
48	<b>0E8</b>	MOV E,8	
49	<b>091</b>	MOV 9,1	
4A	<b>080</b>	MOV 8,0	
4B	<b>C01</b>	GOTO 01	
4C	<b>100</b>	MOVI 0,0	Subroutine „MOVI 0“
4D	<b>101</b>	MOVI 0,1	
4E	<b>102</b>	MOVI 0,2	
4F	<b>103</b>	MOVI 0,3	
50	<b>104</b>	MOVI 0,4	
51	<b>105</b>	MOVI 0,5	
52	<b>F07</b>	RET	

\* Bei diesen Adressen Befehlsänderungen im 2. Teil des Experiments beachten.

Programm starten. Das Display zeigt 00. Zum Eichen des Voltmeters benötigen wir eine 9 V-Batterie, welche schon einige Zeit in Betrieb war. Wir verwenden mit Absicht keine fabrikfrische Batterie, weil solche neue Batterien (trotz entsprechender Aufschrift) meistens 9,6-9,8 V haben, während schon im Einsatz gewesene Batterien ziemlich genau 9 V zeigen, wenn keine Spannungsentnahme vorgenommen wird. Die Batterie wird an den mit  $\oplus$  und  $\ominus$  bezeichneten **Meßleitungen** angeschlossen, (siehe Abbildung). Den Potentiometerknopf drehen wir zurück bis zum linken Anschlag. Im Ohrhörer, bzw. Lautsprecher ist nichts zu hören. Nun wird der Potentiometerknopf in kleinen Positionen langsam nach rechts gedreht. Der Lautsprecher (Ohrhörer) bringt wieder das bekannte knatternde Geräusch. Wir drehen solange nach rechts, bis das Display möglichst genau 90 (9,0 V) anzeigt. Der Computer benötigt für den Meßvorgang ca. 2-3 Sekunden, weshalb nach jeder Potentiometereinstellung der geänderte Anzeigenwert abzuwarten ist.

Vermutlich werden wir feststellen, daß uns die genaue Eichung bei 9 V nicht präzise gelingen wird. Das Display kann schwankende Werte zwischen 8,4 und 9,5 V anzeigen. Dies soll uns zunächst nicht stören.

Mit unserem Digital-Voltmeter können wir jetzt Gleichspannungen **bis zu 9 V** messen. Sicherlich haben wir in einem anderen Elektrogerät 1,5 V Batterien zur Verfügung (z. B. in einem Koffer-Radio), so daß wir bei Verwendung von 2 Batterien 3 V oder bei 3 Batterien 4,5 V usw. ermitteln können. Falls wir das Electronic-Studio „Digital-Technik 2075“ besitzen, kann auch die sehr genau arbeitende 5 V-Buchse des IC-Zählerbausteins zum Eichen und messen verwendet werden.

Unser Digital-Voltmeter hat in der momentanen Ausführungsart eine Meßgenauigkeit von 0,4 bis 0,5 V. Bei Spannungen mit Zwischenwerten ergeben sich die erwähnten Anzeigeschwankungen.

Haben wir jedoch beim Kapitel „Frequenz-Zähler“ festgestellt, daß die Grenz-Frequenz unseres Mikrocomputers über 40 Hz liegt, kann die Meßgenauigkeit unseres Digital-Voltmeters wesentlich verbessert werden. Hierfür müssen die in der Programm-Tabelle mit einem \* gekennzeichneten Befehle gegen die in Klammer angegebenen Befehle getauscht werden. Wir ändern also z. B. bei Adresse 25 den mit „170“ eingegebenen Befehl in „150“ usw.

Nach dieser Programm-Änderung muß das Gerät wie vorstehend beschrieben neu geeicht werden. Die Meßgenauigkeit unseres Digital-Voltmeters liegt nun bereits bei ca. 0,2 V. Es werden sich jedoch bei Spannungen über 6 V immer noch gewisse Anzeigeschwankungen ergeben.

**Achtung! Das Digital-Voltmeter ist nur für Gleichstrom-Messungen bis zu 9 V verwendbar. Wechselstrom-Spannungen werden nicht angezeigt. Der Anschluß von Wechselstrom-Spannungen von über 9 V an den Meßleitungen kann zur Zerstörung von Bauelementen führen.**

#### Wie funktioniert das Digital-Voltmeter?

Den Besitzern des Electronic-Studios „Digital-Technik 2075“ ist das Prinzip der Umsetzung einer analogen Spannung in ein digitales Meßergebnis bekannt.

Bei unserem Experiment erzeugt die astabile Kippstufe eine Frequenz, welche bei geringer Spannung klein ist (langsame Knattern) und bei ansteigender Spannung größer wird (schnelles Knattern). Der Computer zählt die Frequenzen und gibt sie als Spannungsmeßergebnis auf dem Display bekannt.

Beispiel: Die Frequenz beläuft sich auf 20 Hz bei einer angeschlossenen Spannung von 9 V. Die Spannung bei einer Frequenz von nur 10 Hz ergibt sich aus folgender Formel:

$$U = \frac{9 \text{ V}}{20 \text{ Hz}} \times 10 \text{ Hz} = 4,5 \text{ V}$$

oder bei einer Frequenz von 11 Hz:

$$U = \frac{9 \text{ V}}{20 \text{ Hz}} \times 11 \text{ Hz} = 4,95 \text{ V} (\text{aufgerundet } 5 \text{ V}).$$

Bei unserem momentan aufgebauten Experiment ist die Berechnung leider nicht ganz so einfach, weil die höhere oder niedrige Frequenz nicht linear zur angeschlossenen Spannung erzeugt wird. Wenn bei einer Spannung von 9 V die im Beispiel aufgeführten 20 Hz erzeugt werden, müßten bei einem 5 V Anschluß 11 Hz erzeugt werden, um zu einem korrekten Ergebnis zu kommen. Tatsächlich erzeugt unsere astabile Kippstufe bei einer angeschlossenen 5 V Spannung jedoch nicht 11 Hz, sondern 12,5 Hz. Würden wir wieder die gleiche Formel anwenden, kämen wir zu folgendem (falschen) Ergebnis:

$$U = \frac{9 \text{ V}}{20 \text{ Hz}} \times 12,5 \text{ Hz} = 5,6 \text{ V} (\text{richtig wären } 5 \text{ V})$$

Für die Umrechnung der Frequenz in eine Spannung müssen wir bei der zur Verfügung stehenden astabilen Kippstufe eine andere Formel finden. Unter Berücksichtigung der nicht linearen Frequenz-Erzeugung wurde durch Versuche folgende Formel gefunden:

$$U = 0,0067 \times f^2 + 0,3160 \times f$$

Wenn für f die entsprechende Frequenz, z. B. 20 Hz eingesetzt wird, kommen wir zu folgendem Ergebnis:

$$U = 0,0067 \times 20 \times 20 + 0,3160 \times 20 = 9 \text{ V}.$$

Bei 12,5 Hz ergibt sich:

$$U = 0,0067 \times 12,5 \times 12,5 + 0,3160 \times 12,5 = 5 \text{ V}.$$

Da unser Computer keine 12,5 Hz messen kann (sondern entweder 12 oder 13 Hz) ergeben sich bei einer angeschlossenen 5 V Meßspannung schwankende Anzeigenwerte zwischen 4,7 und 5,2 V.

#### Das interessiert den Programmierer

Bis zur Adresse 1B ergibt sich der gleiche Programm-Teil wie beim „Frequenz-Zähler“. Die erzeugte Frequenz wird in die Register A und B übernommen. Diese Werte wandern in die Register 0 und 1 und werden in den Speicherregistern 0 und 1 gespeichert.

Anschließend werden nochmals die Register 0 und 1 mit den Inhalten der Register A und B geladen und der Befehl MULT ausgeführt. Jetzt steht in den Registern 0 bis 5 das Quadrat der Frequenz. Dieses Ergebnis wird mit 67 multipliziert (anstelle 0,0067 in unserer vorerwähnten Formel). Die ersten drei Stellen dieses Ergebnisses (in den Registern 2,3 und 4) wird in die Register D, E und F gespeichert.

In den Registern A und B stehen immer noch die gemessenen Frequenz-Werte, welche jetzt mit 3160 multipliziert werden (anstatt 0,3160 in unserer Formel).

Das Ergebnis wird in den Registern D, E und F hinzugefügt.

Das Endergebnis in den Registern E und F wird in die Register 8 und 9 (beziehungsweise Register 0 und 1) zur Ergebnis-Anzeige umgespeichert.

Der Programm-Ablauf soll nocheinmal übersichtlich dargestellt werden:

1. Frequenz messen (z. B. 14 Hz)
2. Frequenz quadrieren    $14 \times 14 = 196$
3. Ergebnis mit 67 multiplizieren:                                    $196 \times 67 = 13132$
4. Die ersten drei Stellen speichern:                                   = 131
5. Gemessene Frequenz 14 Hz mit 3160 multiplizieren:    $14 \times 3160 = 44240$
6. Die ersten drei Stellen zum Ergebnis von 4. addieren:    $131 + 442 = 573$
7. Die beiden ersten Stellen anzeigen

Wir vergleichen mit der zuvor gefundenen Formel:

$$U = 0,0067 \times 14^2 + 0,3160 \times 14 = 5,737$$

Die Berechnungen des Computers stimmen mit dieser Formel überein. Es wäre sinnlos, das Ergebnis 3-stellig anzugeben, weil die von der astabilen Kippstufe erzeugte Frequenz nicht genau genug ist.

Wird die Frequenz auf 40 Hz (bei 9 V) erhöht, ändert sich die Formel wie folgt:

$$U = 0,0025 \times f^2 + 0,125 \times f$$

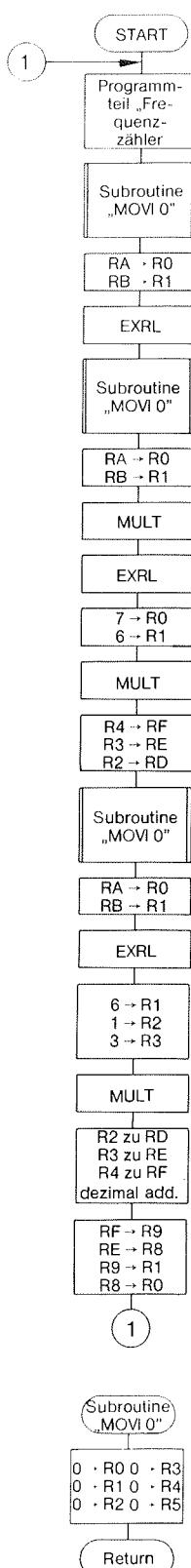
Diese Formel-Änderung wurde bei den in Klammern angegebenen Befehls-Änderungen berücksichtigt.

Eine Ergänzungspackung für exakte Meßergebnisse ist in Vorbereitung. In dieser Packung sind u.a. ein spezieller A/D-Wandler, Temperatur-Fühler usw. enthalten, um sehr genaue Messungen von Spannungen, Temperaturen, Licht usw. durchführen zu können. Unter Einbeziehung dieser Meßwerte die z. B. bei Unterschreitungen oder Überschreitungen vom Computer an die Ausgänge gemeldet werden, ergeben sich viele weitere Einsatzmöglichkeiten des Microtronic-Computersystems.

# Computer gesteuerte Modell-Eisenbahn

## Erste Ausbaustufe

### Programm-Ablaufplan: Digital-Voltmeter.



Die Überwachung und Steuerung einer Modellbahn-Anlage ist eine interessante Anwendung für unseren Computer.

Die Computer-Steuerung bringt gegenüber den bisher gebräuchlichen Relais-Schaltungen große Vorteile: Soll z. B. der Steuerungsablauf geändert werden, ist lediglich eine Änderung des Computer-Programmes notwendig. Eine richtig vorgenommene „Hardware-Verdrahtung“ bleibt unverändert bestehen. Der Computer übernimmt den vollautomatischen Modellbahnbetrieb und durch ein kleines Zusatz-Programm kann diese Automatik auf einen halbautomatischen oder Hand-Betrieb umgeschaltet werden. Man kann auch in einen Automatik-Betrieb eingreifen, indem z. B. eine vom Computer gesteuerte Weichenstellung manuell geändert werden soll. Diese Weichenumschaltung wird durch Computer-Eingabe ange meldet. Der Computer kontrolliert, ob sich ein „Unfall“ ergeben könnte und dirigiert den evtl. „bedrohten“ Zug auf ein anderes Gleis um – die besagte Weiche kann gestellt werden. Falls dies nicht möglich ist, wird die geplante Weichenstellung solange zurückgestellt, bis der gefährdete Zug die evtl. bedrohte Stelle passiert hat. Erst dann wird die manuelle Weichenstellung freigegeben.

Auch die Steuerung sogenannter „Weichen-Straßen“ können Computer-Aufgaben sein.

Manche Modellbahn-Freunde sind der Ansicht, daß lediglich einige Verbindungskabel am Computer angeschlossen werden, um zu einem vollautomatisch gesteuerten Modellbahn Ablauf zu kommen. Nachdem wir uns inzwischen intensiven mit den Möglichkeiten eines Mikrocomputers beschäftigt haben, ist es uns einleuchtend, daß für eine solche Umstellung verschiedene Vorbereitungen zu treffen sind.

### Es ist unbedingt zu beachten:

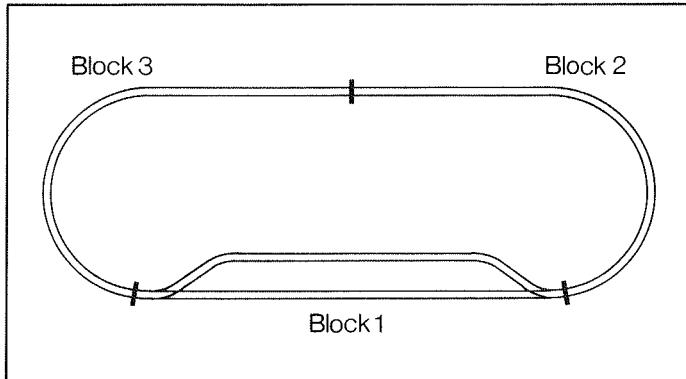
Da der Computer nur mit Gleichspannungen und sehr geringen Strömen arbeitet, dürfen bei den folgenden Experimenten keine Ströme (Spannungen) des Eisenbahn-Transformators an die Ein- oder Ausgänge des Computers gelangen. Weichen, Gleisabschnitte oder andere vom Eisenbahn-Trafo versorgte elektrische Artikel dürfen also niemals mit der Computer-Platine direkt verbunden werden.

Die Gefahr ist groß, daß ein unüberlegter Anschluß zur Zerstörung unseres wertvollen Mikrocomputers führt. So müssen dem Computer über seine Eingänge mitgeteilt werden, welche Gleise besetzt sind, in welche Richtungen die Weichenstellungen verlaufen usw. Hierfür dürfen keine Gleis-Kontakte oder ähnliches am Computer angeschlossen werden. Durch das „Dazwischenschalten“ des speziellen BUSCH-Relais 5964 und der BUSCH-Lichtschranke 5962 stehen uns geeignete Hilfsmittel für die computergesteuerte Modellbahn zur Verfügung.

Zunächst sind Überlegungen notwendig, wie eine Modellbahn-Anlage für die Computer-Steuerung vorzubereiten ist. An einem einfachen Gleisoval mit einem Weichenpaar soll dies demonstriert werden.

Auf diesem Kreis mit Ausweichgleis (Bahnhof) können vollautomatisch und unfallfrei drei Züge verkehren. Während ein Zug am Bahnhof ankommt und dort hält, wird der auf dem zweiten Gleis wartende Zug freigegeben. Der dritte Zug befindet sich auf der „Strecke“.

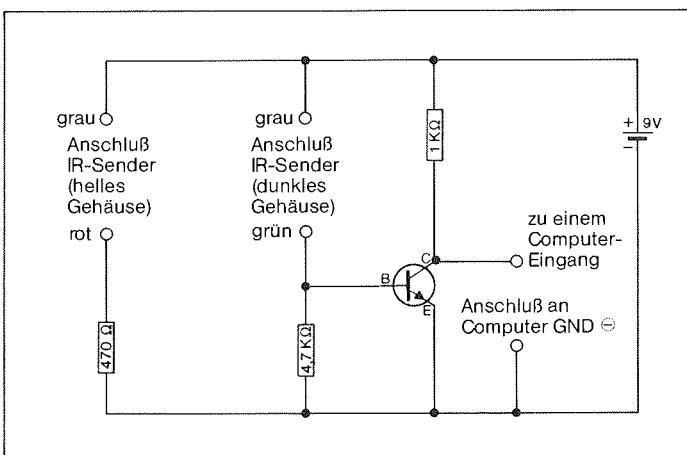
Der Kreis wird, wie auf der Abbildung gezeigt, in drei „Blöcke“ aufgeteilt. Der Computer muß dann prüfen und berechnen, welcher Gleisblock zu sperren oder freizugeben ist und wie die Weichen gestellt werden sollen.



Alle Hersteller von Modell-Eisenbahnen bieten leicht realisierbare Möglichkeiten, die Gleisstrecken durch entsprechende Isolierstücke in einzelne Blockstrecken aufzuteilen.

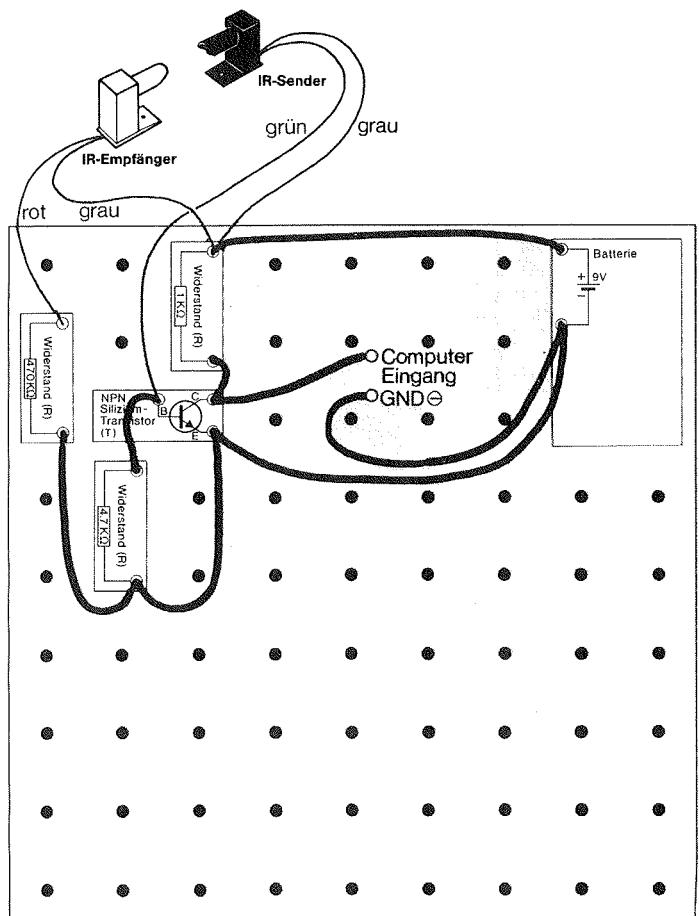
Für die „Überwachung“ der einzelnen Gleisblöcke sind BUSCH-Lichtschranken Nr. 5962 ideal geeignet. Sie sind nicht nur verhältnismäßig preiswert, sondern bieten darüberhinaus die für den Computer erforderliche kontaktlose Datenübertragung. Eine Lichtschranke besteht aus einem kleinen Infrarot-Sender und -Empfänger, welche gegenüberliegend an einem zu überwachenden Gleisabschnitt angebracht werden. Bei Zugdurchfahrt wird ein Signal ausgelöst, welches vom Computer ausgewertet wird. Hierfür ist eine einfache Verstärkerschaltung, bestehend aus einem Transistor und drei Widerständen, erforderlich. Die Besitzer eines BUSCH-Electronic-Studios können die Schaltung durch Verwendung der entsprechenden BUSCH-Steckbausteine leicht aufbauen. Da alle Bauelemente auch einzeln lieferbar sind, ist ein Bezug über die BUSCH-Kundendienststelle möglich. Eine Stückliste finden Sie am Ende dieses Kapitels.

### Schaltplan für IR-Lichtschranken-Verstärkung



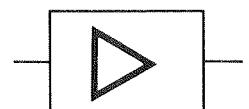
### Aufbauplan für IR-Lichtschranken-Verstärkung

Anstelle des Batterie-Bausteins kann auch das BUSCH-Netzteil 2059 oder BUSCH-Vorschaltgerät 8627 verwendet werden.



Aufbauplan Verstärker-Schaltung für Lichtschranken. Es können mehrere Verstärker-Schaltungen auf einer Steckplatine aufgebaut und von der gleichen Spannungsquelle versorgt werden.

Die kleine Verstärker-Schaltung (gemäß Abbildungen) liefert dem Computer bei Lichtschranken-Unterbrechung (Zugdurchfahrt) ein „1“-Signal (Spannung vorhanden), oder ein „0“-Signal, wenn die Lichtschranke nicht unterbrochen ist. Zur Vereinfachung im Gesamtschaltplan der Modell-Anlage wird die kleine Verstärker-Schaltung durch folgendes Symbol dargestellt:



**Symbol Lichtschranken-Verstärker**

Fahrströme und Weichen werden mit dem BUSCH-Relais 5964 geschaltet. Im Kapitel „Anschluß von Relais“ wurde dargestellt, wie durch eine einfache Transistor-Stufe (1 Widerstand, 1 Transistor) das Relais 5964 durch den Computer angesteuert werden kann.

Für die Stromversorgung des Lichtschranken-Verstärkers und für die einfache Transistorstufe der Relais 5964 wurde in den Aufbauplänen jeweils ein 9 V-Batterieblock vorgesehen. Anstelle des Batterie-Bausteins kann auch das BUSCH-Netzgerät 2059 verwendet werden, welches ebenfalls eine 9 V Gleichspannung (250 mA) abgibt. Für die Relais ist in jedem Fall ein Gleichspannungs-Steuerstrom mit 9 V erforderlich. In keinem Fall darf 9 V Wechselstrom angeschlossen werden. Die Lichtschranken arbeiten in einem Bereich 5 bis 9 V Gleichspannung. Hierfür ist auch das BUSCH-Vorschaltgerät Nr. 8627 geeignet. Dieses Vorschaltgerät bringt beim Anschluß an einen Wechselstrom-Transformator 12 bis 16 V eine 5 V Gleichspannung.

Wir benötigen elektromagnetische Weichen mit Stop-Funktion und **Endabschaltung**. Die Endabschaltung ist erforderlich, weil die Weichen über die Relais einen längeren Schaltimpuls erhalten, der bei Weichen ohne Endabschaltung zur Erhitzung und bei längerem Einsatz zur Zerstörung des elektromagnetischen Antriebs führen kann.

Nun können wir die Verdrahtung unserer zunächst noch einfachen Modell-Anlage vornehmen. Wir benötigen vier Lichtschranken und drei Relais, welche gemäß Gesamtschaltplan zu verdrahten sind. Wir beachten, daß die für die Geradeausstellung der Weichen zuständigen Kabel zur Kontaktstelle 3 des Relais Nr. 1 führen, während die Kabel für die Abbiegestellung an der Kontaktstelle 1 anzuschließen sind. Dies ist wichtig, weil beide Weichen gleichlaufend betätigt werden müssen, d. h., daß beide Weichen entweder auf Geradeausstellung oder auf Abbiegestellung stehen. Nun benötigen wir noch ein Computer-Programm, welches die Kontrolle und automatische Steuerung übernimmt. Hierfür ergeben sich folgende Voraussetzungen:

Ein Zug passiert die Lichtschranke 4 (L4). Der dahinter liegende Gleisblock Nr. 3 muß stromlos geschaltet werden, damit ein evtl. zweiter nachfolgender Zug einem vorausfahrenden Zug nicht zu nahe kommt.

Ein Zug passiert Lichtschranke 1 (L1). Der Gleisblock 2 muß stromlos geschaltet werden. Der Gleisblock 3 wird gleichzeitig wieder freigegeben. Abhängig von der Weichenstellung müssen die Lichtschranken 2 und 3 (L2 und L3) bis zum Eintreffen eines Zuges überwacht werden. Die Lichtschranken 2 und 3 haben folgende Aufgabe:

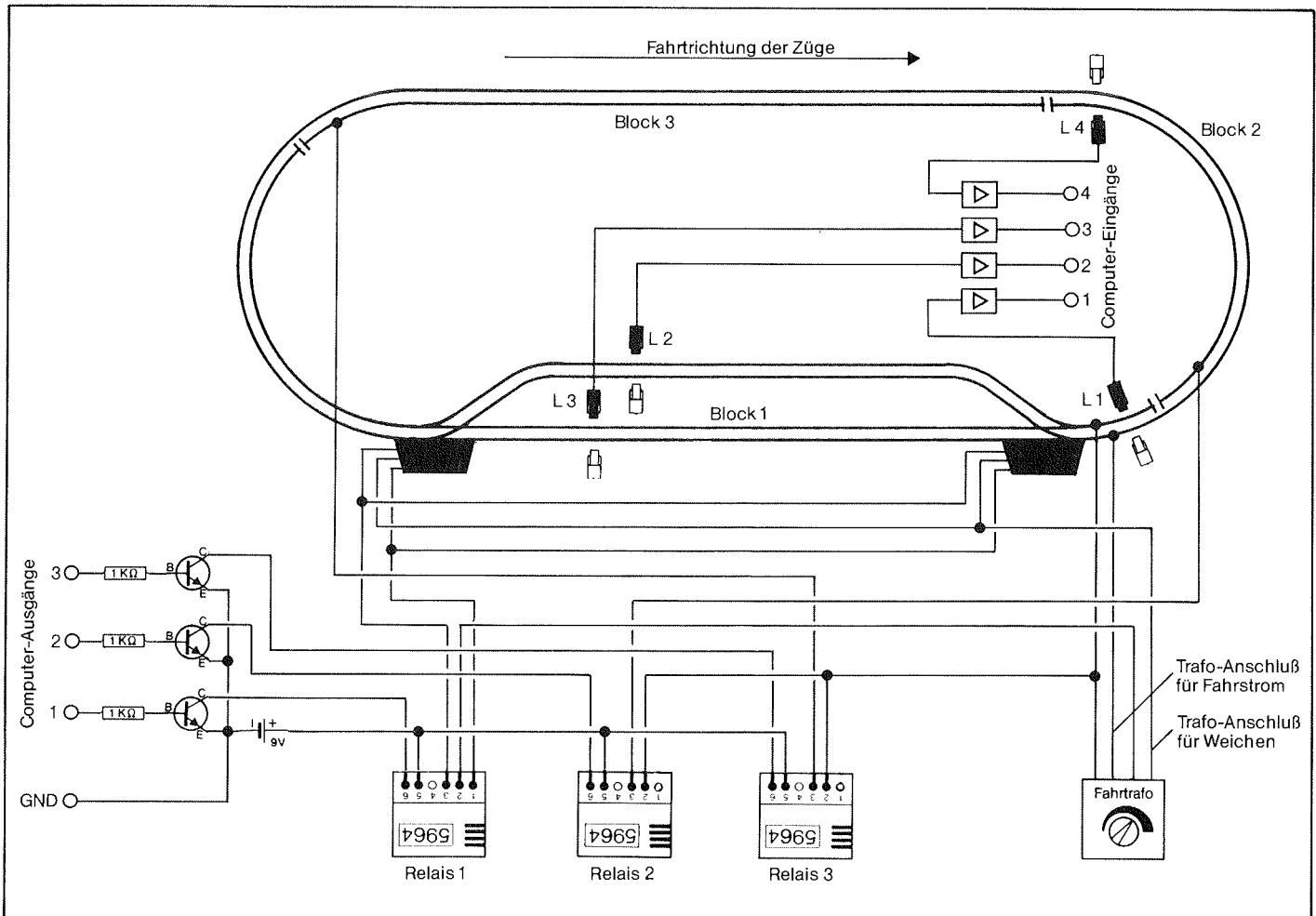
Bei Eintreffen eines Zuges im Bahnhof müssen die beiden Weichen umgeschaltet werden. Bei Verwendung von Stop-Weichen (denkende Weichen) wird hierdurch der Fahrstrom des einfahrenden Zuges abgeschaltet und das gegenüberliegende Gleis freigegeben. Außerdem muß der Gleisblock 2 wieder eingeschaltet werden.

Um diese Überlegungen in ein Computer-Programm einarbeiten zu können, sind eine ganze Reihe sogenannter „logischer Operationen“ (ANDI- bzw. OR-Befehle) erforderlich, um von den 4 Computer-Ein- bzw. Ausgängen immer nur denjenigen anzusprechen, welcher für die Steuerung der jeweiligen Situation erforderlich ist.

Soll z. B. der Zustand der Lichtschranke 4 (Lichtschranke unterbrochen oder Lichtschranke frei, also „1“ oder „0“) übernommen werden, dann müssen alle Werte der 4 angeschlossenen Lichtschranken durch den DIN-Befehl in einem Register gespeichert werden. Durch den Befehl ANDI 8 kann dann der momentane Zustand der Lichtschranke 4 verarbeitet werden. Der Wert der 3 übrigen Eingänge wird kurzzeitig auf 0 gesetzt.

Die Erklärungen in der Programm-Tabelle sollten ausreichend sein, um das Programm zu verstehen. Es arbeitet nacheinander die vorstehend formulierten Bedingungen ab. In Register 0 wird der Wert der Lichtschranken übernommen. In Register 1 wird gespeichert, welche Ausgänge ein- bzw. ausgeschaltet werden. In Register 2 erfolgt eine Zwischenspeicherung falls Lichtschranke L1 unterbrochen wurde, bis die Lichtschranken L2 oder L3 unterbrochen werden. Es muß darauf hingewiesen werden, daß die aus früheren Experimenten evtl. auf der Computer-Platine vorhandene Verbindungsleitung zwischen

### Gesamtschaltplan Modellbahn-Anlage, erste Ausbaustufe



TAKT/CLOCK und einem EINGANG unbedingt entfernt werden muß, weil sich sonst Programm-Störungen ergeben. Wir geben das Programm gemäß Tabelle ein:

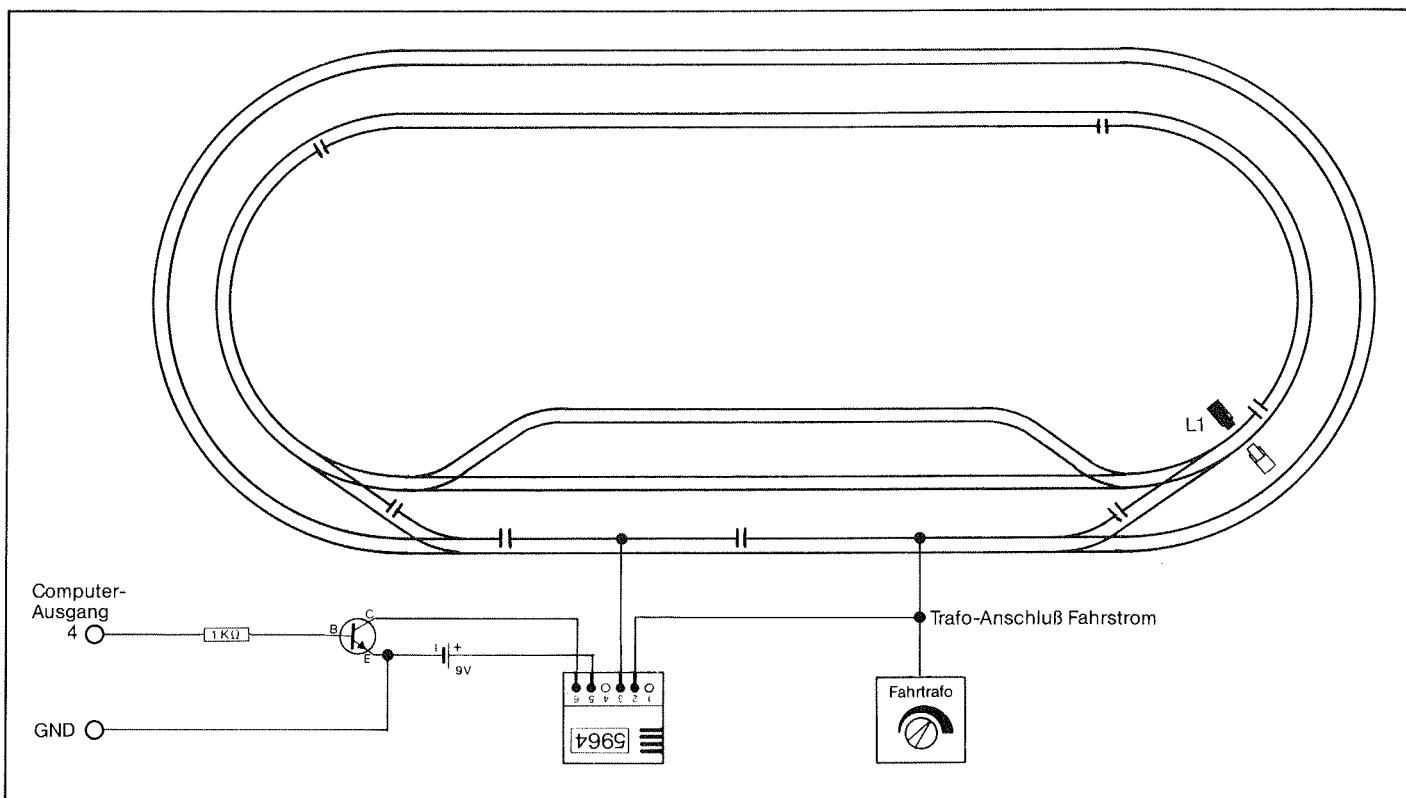
#### Programm: Modellbahn-Steuerung erste Ausbaustufe

Adr.	Eingabe-Befehl	Mnemonic	Erklärungen
00	<b>F08</b>	CLEAR	
01	<b>F02</b>	DISOUT	
02	<b>161</b>	MOVI 6,1	
03	<b>FE1</b>	DOT 1	Wert von R1 auf Ausgänge
04	<b>FD0</b>	DIN 0	Wert von Lichtschranken übernehmen
05	<b>380</b>	ANDI 8,0	Durch ANDI 8,0 nur Wert von L4 übernehmen
06	<b>E08</b>	BRZ 08	Ist L4 gleich 0, dann Sprung zu Adresse 08
07	<b>3B1</b>	ANDI B,1	Durch ANDI B (1011) Block 3 sperren
08	<b>FD0</b>	DIN 0	Wert von Lichtschranken übernehmen
09	<b>310</b>	ANDI 1,0	Durch ANDI 1,0 nur Wert von L1 übernehmen
0A	<b>E0F</b>	BRZ OF	Ist L1 gleich 0, dann Sprung zu Adresse OF
0B	<b>1F2</b>	MOVI F,2	Speichern, daß L1 unterbrochen wurde
0C	<b>3D1</b>	ANDI D,1	Durch ANDI D (1101) Block 2 sperren
0D	<b>14F</b>	MOVI 4,F	Durch ODER-Verknüpfung von 4 (0100) Block 3 freigeben
0E	<b>AF1</b>	OR F,1	
0F	<b>9F2</b>	CMPI F,2	Wurde L1 unterbrochen?
10	<b>E12</b>	BRZ 12	Ja, dann Sprung nach 12
11	<b>C03</b>	GOTO 03	Nein, dann Sprung nach 03
12	<b>013</b>	MOV 1,3	Feststellen, ob die Weichen (Ausgang 1) auf gerade (0) oder Abbiegen (1) stehen.
13	<b>313</b>	ANDI 1,3	
14	<b>E1D</b>	BRZ 1D	
15	<b>FD0</b>	DIN 0	Wert von Lichtschranke übernehmen
16	<b>320</b>	ANDI 2,0	Durch ANDI 2,0 nur Wert von L2 übernehmen
17	<b>E03</b>	BRZ 03	Ist L2 gleich 0, dann Sprung nach 03

18	<b>3E1</b>	ANDI E,1	Durch ANDI E (1110) Weichen auf „gerade“ stellen
19	<b>12F</b>	MOVI 2,F	Durch ODER-Verknüpfung von 2 (0010)
1A	<b>AF1</b>	OR F,1	Block 2 freigeben
1B	<b>102</b>	MOVI 0,2	Speicherung von L1 löschen
1C	<b>C03</b>	GOTO 03	Sprung nach 03
1D	<b>FD0</b>	DIN 0	Wert von Lichtschranken übernehmen
1E	<b>340</b>	ANDI 4,0	Durch ANDI 4,0 nur Wert von L3 übernehmen
1F	<b>E03</b>	BRZ 03	Ist L3 gleich 0, dann Sprung zu Adresse 03
20	<b>11F</b>	MOVI 1,F	Durch ODER-Verknüpfung von 1 (0001)
21	<b>AF1</b>	OR F,1	Weichen auf Abbiegen stellen
22	<b>C19</b>	GOTO 19	Sprung nach 19
23	<b>F01</b>	NOP	
24	<b>F01</b>	NOP	
25	<b>161</b>	MOVI 6,1	Zusatzprogramm für Handbetrieb
26	<b>FE1</b>	DOT 1	
27	<b>FF0</b>	KIN 0	
28	<b>171</b>	MOVI 7,1	
29	<b>FE1</b>	DOT 1	
2A	<b>FF0</b>	KIN 0	
2B	<b>C25</b>	GOTO 25	

Für den vollautomatischen Modellbahn-Betrieb wird das Programm mit HALT – NEXT – 00 – RUN gestartet. Zwei Züge stehen zweckmäßigerweise auf den beiden Gleisen am Bahnhof (Block 1) ein dritter Zug steht gegenüber in Block 3. Logischerweise dürfen keine Lokomotiven oder Wagen auf den Weichen stehen. Nach Einschalten des Fahrstroms stellt sich ein vollautomatischer Modellbahn-Betrieb ein. Das Display ist abgeschaltet, weil der Computer teilweise sehr schnell reagieren (rechnen) muß.

#### Schaltplan zweite Ausbaustufe mit einem zusätzlichen Außenkreis



Ohne Änderung der gesamten elektrischen Verkabelung kann unsere kleine Anlage auch jederzeit im „Handbetrieb“ gefahren werden. In diesem Fall stoppen wir das Programm auf Adresse 25 nach HALT – NEXT – 25 – RUN. Die Gleisblöcke 2 und 3 werden jetzt nicht mehr überwacht, sondern sind ständig eingeschaltet. Für die Weichenstellung kann eine beliebige Zahlen-taste betätigt werden.

Wird die grüne RESET-Taste auf der Computer-Platine betätigt, werden sofort alle Ausgänge abgeschaltet. Damit werden die Gleisblöcke 2 und 3 gesperrt. Die RESET-Taste kann als „Not-Stop-Taste“ verwendet werden.

Schon an dieser kleinen und einfachen Anlage sind die vielfältigen Möglichkeiten einer Computer-Steuerung zu erkennen, die mit bisher üblichen Mitteln (Blockstellen-Steuerung usw.) in dieser Präzision nicht erreichbar ist. Wie bei allen automatisierten Funktionsabläufen ist darauf zu achten, daß z. B. alle Wagen-Kupplungen einwandfrei funktionieren, damit durch stehengebliebene Wagen keine unvorhergesehenen Pannen eintreten.

Unsere kleine Modellbahn-Anlage wird noch interessanter, wenn sie durch einen zusätzlichen Außenkreis vervollständigt wird.

Mit zusätzlichem Gleismaterial und zwei weiteren Weichenpaaren wird die Anlage gemäß Abbildung aufgebaut. Der Innenkreis entspricht unverändert dem bisherigen Aufbau. Alle Lichtschranken- und Relais-Anschlüsse bleiben bestehen. Es ist lediglich darauf zu achten, die Lichtschranke L1 rechts neben der neu eingefügten Weiche angeordnet wird.

Da wir noch einen Computer-Ausgang frei haben, können wir auf dem neuen äußeren Kreis im Bahnhofsbereich eine „Haltestelle“ einrichten, die durch ein weiteres Relais über den 4. Computer-Ausgang angesteuert wird. Dieser Außenkreis ist für Fernzüge gedacht, welche nur gelegentlich für eine gewisse Zeit am Bahnhof halten und dann automatisch weiterfahren. Um dies zu erreichen, wird das bereits eingegebene Programm ab Adresse 2C vervollständigt. Zusatz-Programm-Eingabe nach HALT – NEXT – 2C:

### Zusatzprogramm: Modellbahnsteuerung 2. Ausbaustufe

Adresse	Eingabe-Befehl	Mnemonic	Erklärungen
2C	<b>516</b>	ADDI 1,6	Haltezeit im Bahnhof Außenkreis
2D	<b>FB7</b>	ADC 7	
2E	<b>977</b>	CMPI 7,7	
2F	<b>D32</b>	BRC 32	
30	<b>371</b>	ANDI 7,1	
31	<b>C08</b>	GOTO 08	
32	<b>188</b>	MOVI 8,8	
33	<b>A81</b>	OR 8,1	
34	<b>C08</b>	GOTO 08	
35	<b>FD0</b>	DIN 0	Haltezeit vor Bahnhof Innenkreis
36	<b>549</b>	ADDI 4,9	
37	<b>FBA</b>	ADC A	
38	<b>E3A</b>	BRZ 3A	
39	<b>F07</b>	RET	
3A	<b>12F</b>	MOVI 2,F	
3B	<b>AF1</b>	OR F,1	
3C	<b>C03</b>	GOTO 03	

Bevor wir das Programm starten, sind beim ersten Programm-Teil noch folgende Änderungen durchzuführen: Mit HALT – NEXT – 06 die erste Befehls-Änderung vornehmen und dann mit NEXT-Taste zur nächsten zu ändernden Adresse weitergehen und auch nach der letzten Befehls-Änderung nicht vergessen, nochmals die NEXT-Taste zu betätigen:

Bei Adresse 06 Befehls-Eingabe ändern in **E2C** (BRZ 2C)

Bei Adresse 15 Befehls-Eingabe ändern in **B35** (CALL 35)

Bei Adresse 1D Befehls-Eingabe ändern in **B35** (CALL 35)

Alle übrigen Funktionen bleiben unverändert. Programm-Start mit HALT – NEXT – 00 – RUN.

Auf dem Außenkreis kann ein vierter Zug entweder mit dem gleichen Fahrstrom-Transformator mit dem auch bereits die drei übrigen Züge fahren, oder mit einem zusätzlichen Fahrstrom-Trafo in Betrieb genommen werden. Auf unserer Anlage wird sich nun „ein emsiges Treiben“ entwickeln. Da die Bahn auf dem Außenkreis nur gelegentlich hält, ergeben sich durch die vier gleichzeitig verkehrenden Züge für einen Laien, welcher nicht weiß, daß es sich um eine Computer gesteuerte Anlage handelt, ein faszinierender Eindruck.

Die Haltezeit des D-Zugs auf dem Außenkreis ist variabel. Sie kann verkürzt werden, wenn der Befehl auf Adresse 2C durch die Eingabe 526 oder 536 oder 546 usw. geändert wird.

Wenn alle vier Züge in der gleichen Richtung fahren, kann durch manuelle Betätigung bzw. durch die üblichen Fernsteuer-Weichenschalter ein Zug vom Innenkreis zum Außenkreis wechseln. Sobald ein Zug den Innenkreis verläßt, bleibt ein unmittelbar folgender Zug für ca. 5 Sekunden im Gleisblock 2 stehen, wodurch sich genügend Zeit für die Weichenrückstellung ergibt. Auch diese Zeit kann verlängert werden, wenn der Befehl auf Adresse 36 durch die Eingabe 539 oder 529 oder 519 verändert wird. Da auf dem Außenkreis keine Lichtschranken-Überwachung vorhanden ist, sollte dort jeweils nur ein Zug verkehren, d. h. daß bei einem Wechsel vom Innenkreis zum Außenkreis einer der beiden dort vorübergehend verkehrenden Züge alsbald wieder zum Innenkreis zurück wechseln sollte.

Das Zusatz-Programm beinhaltet von Adresse 2C bis 34 einen zweistelligen Zähler, durch welchen der Computer-Ausgang 4 je nach Zählerstand ein- oder ausgeschaltet wird. Da es sich um einen „Zufalls-Zähler“ handelt, hält der auf dem Außenkreis verkehrende Zug in unregelmäßiger Folge. Ab Adresse 35 folgt ebenfalls ein Zähler, welcher den Block 2 (Computer-Ausgang 2) nach ca. 5 Sekunden freigibt und nicht wie normalerweise nach Durchfahrt eines Zuges durch die Lichtschranke L2 oder L3. Somit wurden 2 programmierte Zeitschalter in den Programm-Ablauf integriert.

Selbstverständlich können wir unsere Anlage durch nicht vom Computer überwachte Abstellgleise usw. weiter ausbauen. Auch an den Relais Nr. 2 und 3 sind noch jeweils eine Anschlußbuchse (Nr. 1) frei, an welchen z. B. weitere Relais angeschlossen werden können, um damit eine Warnblinkanlage am Bahnübergang oder Beleuchtungen innerhalb der Modellstadt einzuschalten. Am Relais Nr. 1 könnten zusätzliche Signale (gekoppelt mit der Weichenstellung) angeschlossen werden. Am Computer selbst haben wir alle Ein- und Ausgänge belegt, weshalb für weitere Computer-Funktionen zusätzliche Ergänzungs-Elemente notwendig sind.

BUSCH wird speziell für Modell-Eisenbahner zusätzliche Speicher-Bausteine entwickeln, so daß für einen weiteren Ausbau die jetzt vorhandenen vier Ein- und Ausgänge vervielfacht werden. Interessierte Modell-Eisenbahner beachten bitte die Ankündigungen im neuen BUSCH-Modellbahn-Katalog, welche ab Spätsommer 1982 zur Verfügung steht. Sie werden staunen, was aus dem Microtronic-Computersystem noch „herauszuholen“ ist.

## **Stückliste: Zusätzliche Bauelemente für Modellbahn-Steuerung**

Unter Berücksichtigung der in der Packung 2090 enthaltenen Elektronik-Bausteine werden für den Aufbau der Modellbahn-Steuerung noch folgende Einzelteile benötigt:

4	Bausteine Widerstand 470 Ohm	Nr. 20684
7	Bausteine Widerstand 1 k Ohm	Nr. 20685
2	Bausteine Widerstand 4,7 K Ohm	Nr. 20687
7	Bausteine NPN-Transistoren	Nr. 20611
1	Steckplatine (für Bausteine)	Nr. 20782
1	Beutel Kabelabschnitte in verschiedenen Längen	Nr. 20785
1	Beutel 48 gelbe Plastikstecker	Nr. 20795

Diese Teile können Sie bei Ihrem Microtronic-Fachhändler oder direkt bei uns (gegen Voreinsendung des Betrags auf Postscheckkonto) bestellen. Bitte, verwenden Sie das beigelegte Bestellformular, in welchem Sie alle erforderlichen Angaben finden. Ggf. neues Bestellformular kostenlos bei uns anfordern.

Die übrigen notwendigen Teile:

4	IR-Lichtschranken	Nr. 5962
4	Spezial-Relais	Nr. 5964

sollten in den Modellbahn-Abteilungen guter Spielwaren-Fachgeschäfte vorrätig sein.

## **Kann das Microtronic-Computersystem noch mehr?**

Es ist unmöglich, für alle Anwendungsfälle ein Beispiel in diesem Anleitungsbuch aufzuführen. Das Buch käme nie zu einem Ende, weil die Verwendungsmöglichkeiten eines derartigen Mikrocomputers nicht übersehbar sind.

Deshalb beginnt jetzt der spannendste Teil der Beschäftigung mit dem Microtronic-Computersystem:

Wir überlegen uns, welche Aufgaben der Computer übernehmen kann. Wir bringen unsere Überlegungen in die für den Computer durchführbare Reihenfolge und wir entwickeln die hierfür erforderlichen Programme. Für viele Anwendungsfälle können geänderte Programm-Teile der Anleitungsbücher verwendet werden. Bei intensiver Beschäftigung werden wir jedoch noch unendliche Programmierungsmöglichkeiten entdecken. Es liegt an uns, diese Möglichkeiten wahrzunehmen. Wir werden feststellen, daß unser Mikrocomputer nicht nur eine ernsthafte und befriedigende Freizeitbeschäftigung vermittelt, sondern daß er auch für viele Verwendungsmöglichkeiten in der Praxis eingesetzt werden kann. BUSCH bringt hierfür im Laufe der Zeit weitere Ergänzungen auf den Markt, wodurch das Microtronic-System dem jeweiligen Stand der Technik angepaßt wird.

## **Microtonic-Programmier-Wettbewerb**

Sinn des Microtonic-Computersystems ist es, die Möglichkeiten der Mikrocomputer-Technik zu beherrschen und zu immer neuen Überlegungen und Programmierungen zu kommen.

Um den Erfahrungsaustausch zu koordinieren ist es geplant, einen Wettbewerb durchzuführen unter dem Motto:

### **„Die interessantesten Programm-Vorschläge für das Microtonic-Computersystem“**

Einsendungen sind jederzeit möglich. Die von BUSCH prämierten Programm-Vorschläge werden in künftigen Microtonic-Programmbüchern veröffentlicht. Jedes zur Veröffentlichung kommende Programm wird durch Auszahlung einer Prämie von DM 100,- honoriert.

## **Teilnahmebedingungen für den BUSCH-Microtonic-Programm-Wettbewerb**

Teilnahmeberechtigt sind alle Benutzer eines Microtonic-Computersystems, ausgenommen Belegschaftsmitglieder der Firma BUSCH GmbH.

Die einzusendenden Programm-Vorschläge müssen neu sein, d. h. sie sollten keine geänderten Variationen bereits veröffentlichter Programme darstellen und sie sollten sich im Rahmen der folgenden 3 Kategorien bewegen:

### **1. Programm-Vorschläge, welche mit dem Microtonic-Computersystem ohne weiteres Zubehör durchführbar sind.**

Für die Funktion dieser Programme darf nur der Microtonic-Computer und die im Kasten 2090 enthaltenen Ergänzungsteile verwendet werden.

### **2. Programm-Vorschläge unter Verwendung zusätzlicher BUSCH-Electronic-Bauelemente.**

Für diese Programm-Funktionen können alle Einzelteile der BUSCH-Electronic-Studios, BUSCH-Schalt-Relais und die weiter von BUSCH auf den Markt kommenden Ergänzungselemente eingesetzt werden.

### **3. Programm-Vorschläge unter Verwendung allgemein käuflicher Bauelemente.**

Für diese Programm-Funktionen können allgemeine, serienmäßige und überall erhältliche Bauelemente einbezogen werden. Der Kaufpreis (allgemeiner Ladenpreis) der zusätzlich notwendigen Bauteile sollte den Betrag DM 80,- nicht überschreiten.

Die für eine Wettbewerbsbeteiligung eingesandten Programme sollen in der Art der Microtonic-Programm-Listen aufgemacht sein.

Erforderlich ist eine kurze Funktionsbeschreibung, notwendige Programm-Erklärungen und bei größeren Programmen ein Programm-Ablaufplan.

Werden zusätzliche elektronische Bauelemente benötigt, ist die Beifügung eines Schalt- und Aufbauplans notwendig.

Bitte, adressieren Sie Ihre Programm-Vorschläge an:

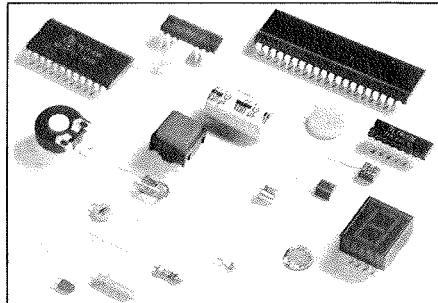
BUSCH GmbH & Co. KG  
Abteilung Microtonic-Wettbewerb  
Postfach 1360  
6806 Vierheim/W-Germany

Konzeption und Logik:

# Das elektronische Experimentier-System von

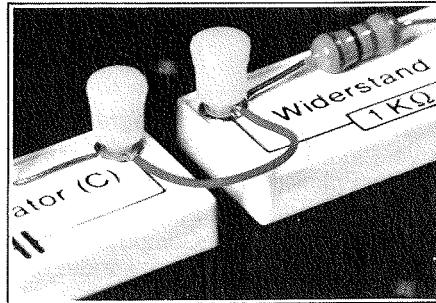


In Zusammenarbeit mit dem Elektronik-Magazin



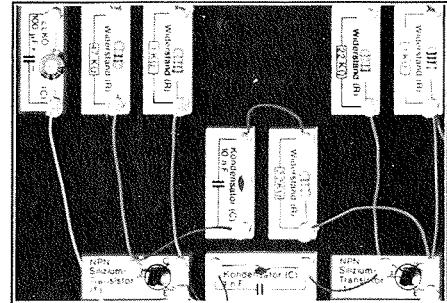
## Moderne Industriebauteile...

Vom einfachen Widerstand über Kondensatoren, Dioden, Fotozellen, Transistoren, Lautsprecher usw. bis zum IC und Mikroprozessor.



## ... auf BUSCH-Elektronik-Bausteine montiert

Einbau fertig und funktions geprüft. Durch Elementbezeichnung und Schaltbild für jeden verständlich.



## Perfekte Experimentier-Technik...

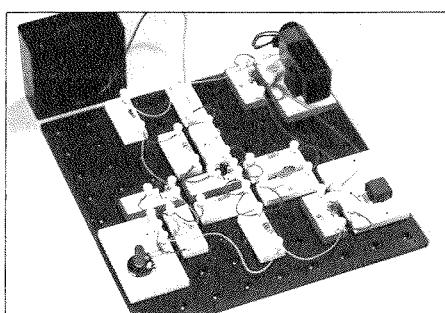
Für den Schaltungsaufbau die Bausteine auf Steckplatine aufsetzen und mit der geschützten BUSCH-Leitungs-Klemmtechnik absolut kontaktsicher verdrahten

## Compact-Studio 2060

### Grundkasten

ab 10 – 12 Jahre

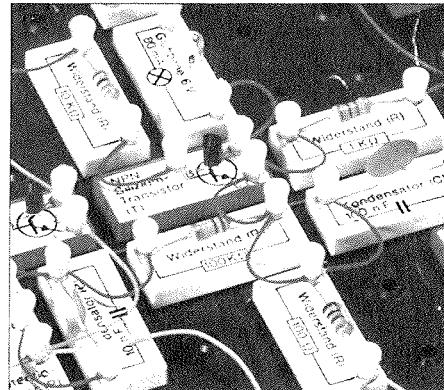
Erweiterungsfähig mit den Electronic-Studios 2061, 2070, 2090.



Ohne Vorkenntnisse sofort experimentieren! Die preiswerte Startpackung für solche, die erst einmal probieren und später weiter ergänzen möchten.

Ein ausführliches Anleitungsbuch mit über 120 Abbildungen und Schaltplänen, zeigt für jeden verständlich den Aufbau und die Funktion von 40 interessanten Experimenten und Geräteschaltungen.

**Geräteschaltungen:** Vom einfachen Stromkreis bis zur Lichtorgel-Prinzipschaltung. U. a. Regen-Warnanlage, Spannungsprüfgerät, Alarm-Anlagen, Dimmerschaltung, automatische Zeit- und Verzögerungsschaltungen, Blinklichtschaltungen, elektronisches Martinshorn, Sirenenheulen, Tongeneratoren, akustischer Trittmelder, kapazitiver Annäherungsmelder, ferngesteuertes elektronisches Relais, NF-Tonverstärker, Morsezeichengeber, elektronische Mini-Orgel.



**Inhalt:** Die Packung enthält alles, was für den Anfang notwendig ist wie z. B., Mini-Lautsprecher-Box, Elektronik Bausteine mit Transistoren, Potentiometer, Kondensatoren, Widerstände, Taster, Glühlampe, Batteriehalter, Steckplatine, Kabel, abisolierte Kabelabschnitte in verschiedenen Längen, Klemmstecker usw. (ca. 100 Einzelteile).

## Ergänzungspackung Netzgerät 2059



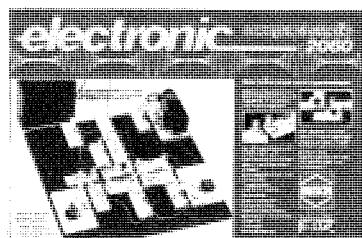
Für sämtliche Electronic-Studios geeignet.



Für alle, die länger experimentieren oder auf gebaute Geräteschaltungen (z. B. Wecker-Radios) im Dauereinsatz verwenden, wird die Anschaffung des speziellen Netzteils 2059 empfohlen. Der Steckbaustein des Netzteils wird in gleicher Weise wie der sonst übliche Batterie-Baustein innerhalb der Schaltungen eingesetzt. Das Gerät spart Batterien und sorgt immer für die richtig dosierte Spannung.

Das Netzteil 2059 ist ein kompakter, doppelt isolierter Sicherheitstransformator. Die integrierte elektronische Stabilisierung sorgt für gleichmäßige Spannungsabgabe mit Brummsiebung (wichtig bei Radioversuchen).

Eingang (primär) 220 V Wechselstrom. Ausgang (sekundär) 9 V Gleichstrom, maximal 200 mA. Kurzschlußsicher nach VDE-Richtlinien geprüft.



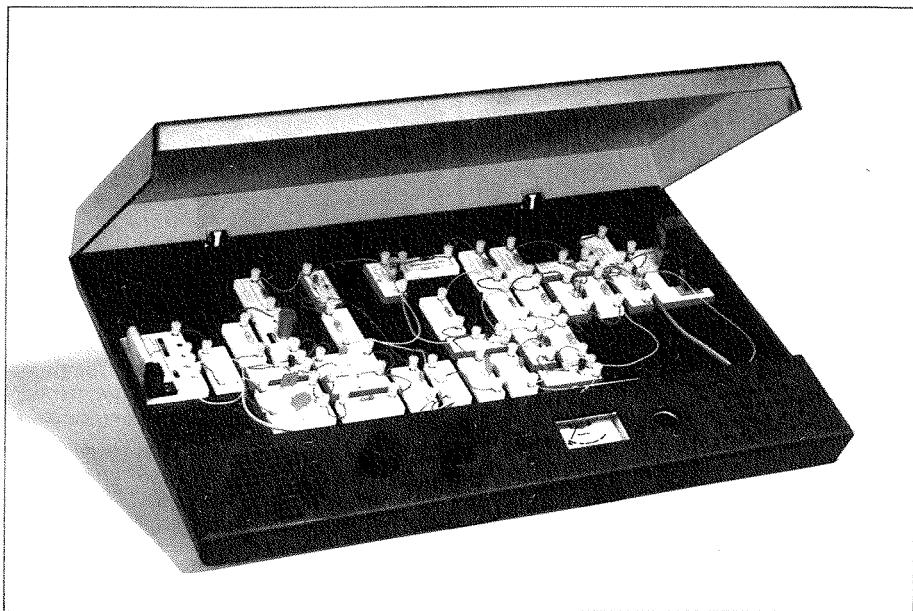
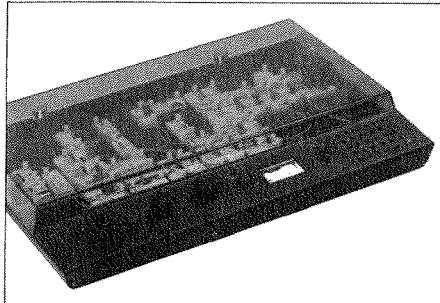
# Studio-Center

## 2070

Grundkasten  
ab 12 – 14 Jahre

Erweiterungsfähig mit den Electronic-Studios 2060, 2072, 2075 und 2090.

Das große Studio-Center 2070 ist Mittelpunkt der BUSCH-Electronic-Studio-Serie. Es bietet Rundum – Experimentiermöglichkeiten in allen Bereichen der modernen Elektronik. Durch Schaltungsaufbau im Studio-Gehäuse entstehen über 130 realistisch funktionierende Geräte. Umfangreiches Anleitungsbuch mit ca. 370 Abbildungen, für jedermann verständliche Aufbaupläne und professionelle Schaltplandarstellungen. Eine exzellente Sache, auch für solche, die nicht nur experimentieren möchten.



**Geräteschaltungen** z. B.: Mittelwellen-Radios mit Gegenaktendstufe, Langwellen (LW)-Empfänger, Kurzwellen (KW)-Radio, UKW-Rundfunkempfänger, 1-Kanal-Lichtorgel, Beschleunigungs- und Phon-Meßgeräte, Leistungs- und Magnetfeldsuchgeräte, hochempfindliche Telefon-Mithörschaltung und Abhöranlage, Sender-Versuche, elektronisches Klavier und Hawaii-Gitarre, automatische Telefonwähl scheibe, Rhythmus-Gerät, Auto-Alarmanlage, Reaktionszeit-Meßgerät, digitaler Zähler, elektronischer Würfel, Hörfähigkeitstester, akustisch ferngesteuerte Schalter, Schußdedektor, Reaktionszeit-Testgerät, elektronisches Roulette, Metallsuchgerät, automatische Warnblinkanlage, Lichtschranke mit akustischem Daueralarm, elektronische Nervensäge und vieles mehr. Außerdem sind sämtliche bei 2060 und 2065 aufgeführten Geräteschaltungen möglich.

**Inhalt:** Electronic-Studio-Center mit rauchglasfarbiger Abdeckhaube und Armaturenboard mit eingebautem Lautsprecher, Potentiometer, Drehkondensator, Schiebeschalter, Universal-Meßgerät und Phono-Normbuchse. Außerdem eine reichhaltige Auswahl von Elektronik-Bausteinen mit Transistoren, Radio-IC (integrierter Schaltkreis), Diode und Leuchtdiode, Elektrolyt- und Scheiben kondensatoren, Fotowiderstand, Ferritantennen, Drosselspule, Potentiometer, Widerstände, Taster, Ohrhörer, Glühlampe, Drähte, Kabel, Kabelabschnitte und Klemmstecker.



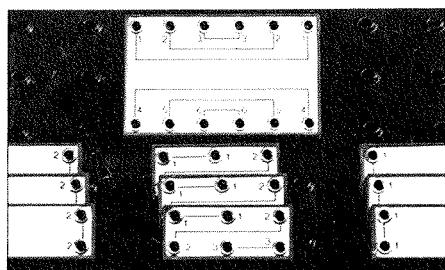
## Ergänzungspackung

### 2079

#### Steckbausteine zum Selbstbestücken.

Möglichkeiten des unendlichen Weiterexperimentierens: ohne zu löten beliebige elektronische Bauelemente auf den Bausteinen befestigen.

**Inhalt:** 10 Bausteine und 45 Klemmstecker. Für die leichte Selbstmontage von 2-, 3- oder 6-poligen Bauelementen wie Widerstände, Kondensatoren, Transistoren, Thyristoren, Spulen, Transformatoren, Schaltrelais usw.



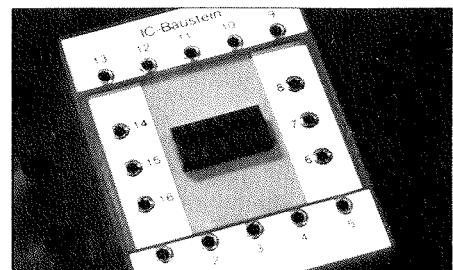
## Ergänzungspackung

### 2089

#### IC-Fassungen zum Selbstbestücken.

Wenn durch die Experimente mit den Electronic-Studios 2072 und 2075 und vor allem beim Mikro-Computer 2090 erste IC-Erfahrungen vorhanden sind, ergeben sich mit diesen Selbstbestückbaren IC-Bausteinen unendliche Erweiterungsmöglichkeiten.

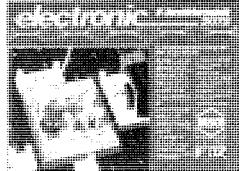
**Inhalt:** 2 Bausteine zum Einstechen beliebiger IC's bis 16 Anschluß-Pins, sowie 40 Klemmstecker. Ausführliche Anleitung mit Hinweisen auf die gebräuchlichsten IC-Typen, Anwendungsbeispielen mit Schalt- und Aufbau-Plänen.



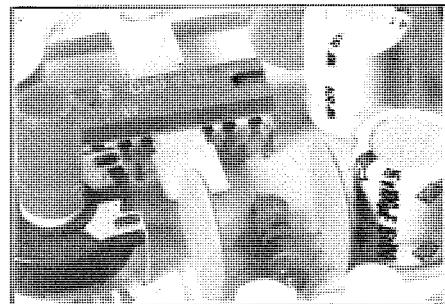
## Ergänzungspackung IC-Verstärkertechnik

2072

ab 12 Jahre



Ideal zur Ergänzung  
der Electronic-Studios 2065, 2070 und 2090.



Dieser Zusatzkasten „IC-Verstärkertechnik“  
bringt Power und den Hifi-Super-Sound!

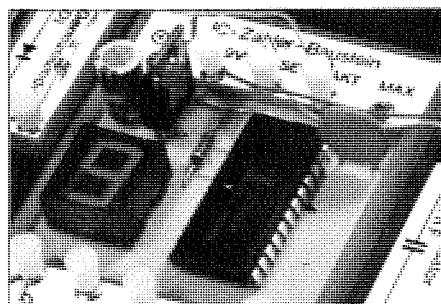
**Geräteschaltungen:** Das ausführliche Anleitungsbuch mit über 50 Abbildungen und Aufbauplänen zeigt 30 tonerzeugende Erweiterungsschaltungen für Rundfunkempfänger mit HiFi-Klang, Phono-Verstärker mit Power-Endstufe, elektronische Orgel, Spinetts und Hawaii-Gitarre, Rhythmusgerät, Wechselsprechsanlagen, HiFi-Mono-Verstärker mit Höhen- und Baß-Regelung, HiFi-Stereo-Verstärker, Telefon-Verstärker, Vibrato-Orgel mit Lichtsteuerung, Hall-Effekte, elektronischer Pausenzeichengeber, Tonübertragung durch Licht usw. Ein echter 2-Kanal-HiFi-Stereo-Verstärker ist durch Verwendung von 2 x 2072 möglich.

**Inhalt:** Auf einem Spezial-Baustein mit einer gearbeiteter Platine ist ein hochwertiger NF-Leistungsverstärker aufgebaut, bestehend aus einem Verstärker-IC, mehreren Elkos, Scheiben kondensatoren, Widerständen und Trimm potentiometern zum Abstimmen. Mit der u. a. beigegebenen Anschlußbuchse ist auch das Überspielen auf große Außenlautsprecher (HiFi-Boxen) möglich.

## Digital-Technik 2075

Grundkasten  
ab 14 Jahre

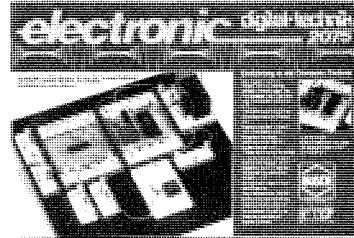
Erweiterungsfähig mit den Electronic-Studios  
2070 und 2090.



Das Studio „Digital-Technik“ erklärt die Grundlagen der Rechner- und Computer-Technik. Selbstständiger Experimentierkasten und gleichzeitig Ergänzungskasten.

Kernstück ist ein IC-Zähler-Baustein mit 7-Segment-Leuchtzifferanzeige und einem integrierten Schaltkreis mit über 350 Transistor-Funktionen. Dieser IC (ein sog. 4-Bit-Counter-Latch-Decoder) enthält außer einem Zähler, einen Speicher und einen Decoder zur Ansteuerung der Leuchtanzeige.

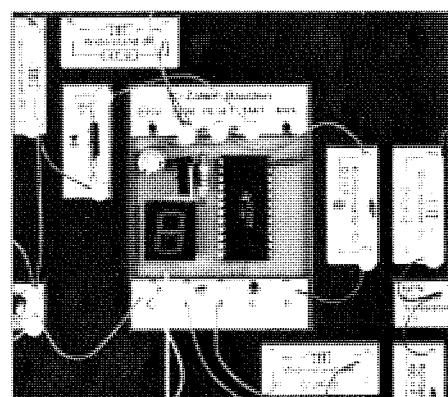
Die Digital-Elektronik erobert sich immer weitere Anwendungsgebiete. Bekannteste Beispiele für diese neue Technik sind Computer und Taschenrechner. Die Digital-Technik wird jedoch auch in der Unterhaltungselektronik bei Fernsehgeräten und großen Receivern genauso eingesetzt wie bei Digitaluhren oder zur Programmsteuerung von Waschmaschinen. Das neueste technische Wunder, der Mikroprozessor (siehe 2090) arbeitet ebenfalls nach den Grundregeln der Digital-Technik.



**Geräteschaltungen:** Das ausführliche Anleitungsbuch mit über 100 Abbildungen und Schaltplänen bringt 30 hochinteressante Experimente wie z. B. Prinzip der Digitaluhr, programmierbarer Zähler, elektronisches Roulette mit Leuchtanzeige, Memory-Schaltung, digitaler Speicher, elektronischer Rechner, Digital-Stoppuhr, Opto-elektronischer Addierer, Frequenz-Meßgerät, Prinzip des Synthesizers, Opto-akustischer-Zähler, Digital-Analog-Konverter, Digital-Volt-Meter, elektronische Spieluhr usw.

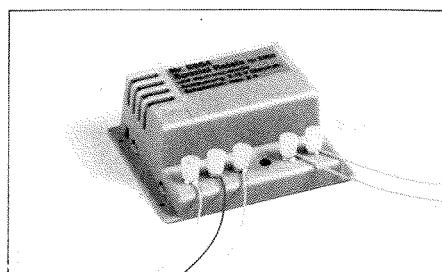
Das Studio Digital-Technik ist gleichzeitig empfehlenswerte Vorstufe zum Microtronic-Computer-System 2090.

**Inhalt:** Über 60 Einzelteile u. a. hochwertiger IC-Zähler-Baustein mit Leuchtanzeige, IC-Gatter-Baustein sowie Widerstände, Elkos, Kondensatoren, Leuchtdiode, Taster, Batteriehalter, Kabel und Klemmstecker.



## Schwachstrom- Spezial-Relais 5964

Dieses Relais kann durch einen minimalen Steuерstrom (8-16 V, 50 mA) der Electronic-Studio-Schaltungen oder durch den Computer 2090 angesteuert werden und schaltet (in Verbindung z. B. mit einem Eisenbahn-Trafo) bis zu 8 A bei 16 V. Ideal bei Experimenten mit Licht-, Schall- und zeit-abhängigen Schaltungen mit der Modelleisenbahn. Auch in Verbindung mit der BUSCH IR-Lichtschranke/Zeitschalter 5961 verwendbar (siehe BUSCH Modellbahn-Zubehör-Katalog).



## Netzstrom-Schaltgerät 2087

für alle Electronic-Studios und Computer 2090.

Das Schaltgerät wird durch die mit den Electronic-Studios aufgebauten Schaltungen mit der Experimentier-Niederspannung (6-9 V) angesteuert und löst über ein Relais Schaltvorgänge im Lichtstromnetz (220 V max. 1000 Watt) aus. Z. B. Ein- und Ausschaltung von Radios, Tonbandgeräten, Warnanlagen, Heizlüftern usw. Die Starkstromseite ist völlig gekapselt und von der Schwachstromseite getrennt, daher auch von Laien und Kindern absolut gefahrlos zu bedienen.

Nach VDE-Vorschriften geprüft. Ausführliche Anleitung mit Schaltbeispielen.



# 2090

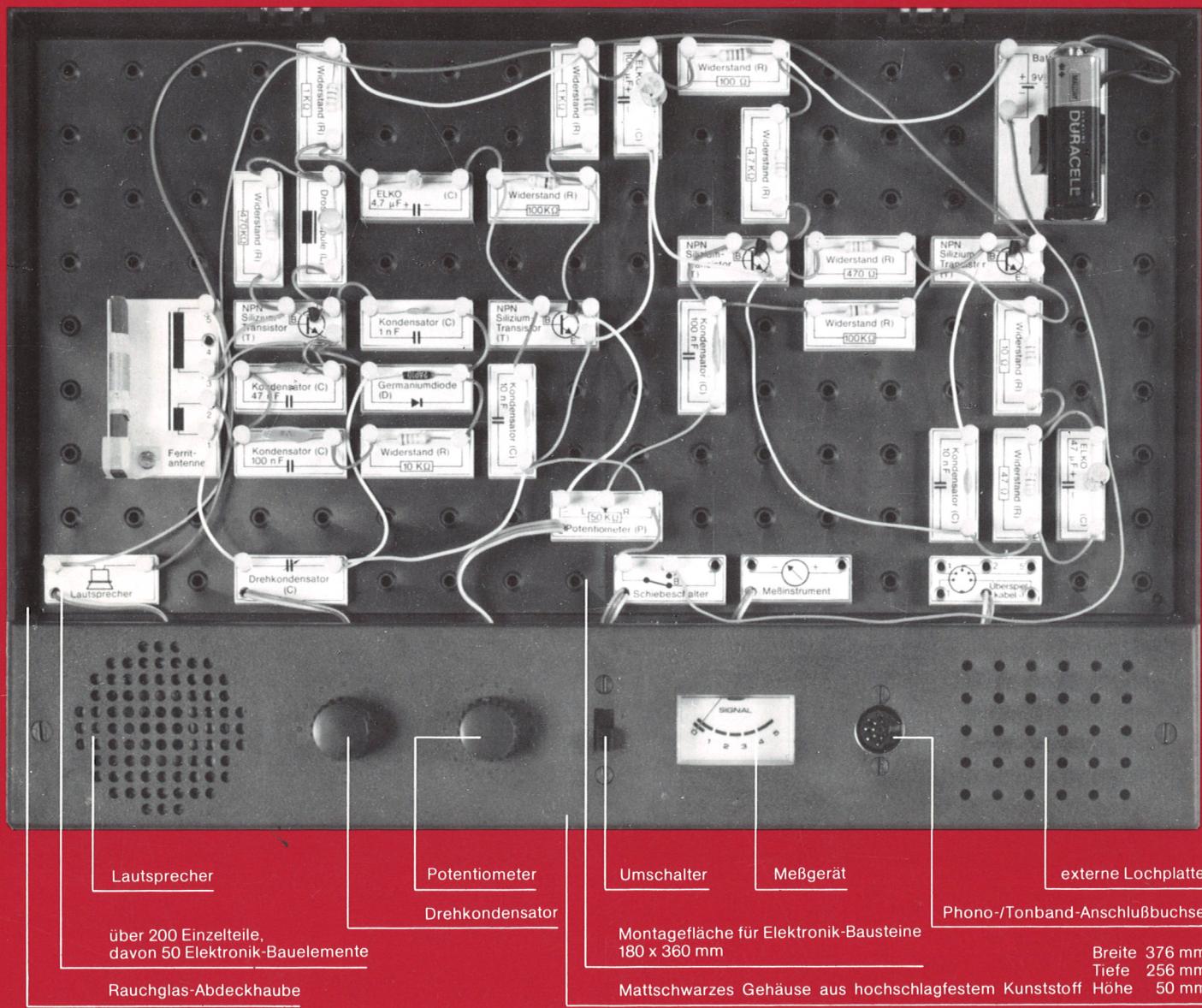


In Zusammenarbeit mit  
dem Elektronik-Magazin



## Perfekt weiter experimentieren . . .

BUSCH Electronic-Studios sind die ideale Ergänzung für das Microtronic-Computersystem. Z. B. das Studio-Center 2070 mit ca. 130 interessanten Experimenten und Geräteschaltungen im Gesamtbereich der modernen Elektronik.



Breite 376 mm  
Tiefe 256 mm

Höhe 50 mm

Montagefläche für Elektronik-Bausteine

180 x 360 mm

über 200 Einzelteile,  
davon 50 Elektronik-Bauelemente

Rauchglas-Abdeckhaube