

Programm zur Decodierung der „Eurosignal“ -Tonfolge

aus dem RPB-Buch Nr.173

„Anwendungsbeispiele für de Mikroprozessor 6502“

von Herwig Feichtinger DC1YB

ISBN 3-7723-1731-6

Seiten 141-149:

Kapitel 7: Vollständige Anwendungsprogramme

Die im letzten Kapitel besprochenen Routinen stellen keine Kompletprogramme für Einplatinencomputer dar, sondern dienen nur als Unterprogramme für bestimmte Teilaufgaben des Systems. Die im folgenden besprochenen Programme sind dagegen jeweils die komplette Betriebssoftware zur Lösung einer bestimmten Aufgabenstellung. Sie unterscheiden sich von den bisher aufgelisteten Teilprogrammen in zwei Punkten:

1. Im Assembler-Listing wird der Reset-Vektor (0FFC im EMUF) auf die Startadresse des Applikationsprogramms gesetzt. Damit wird der richtige Programmbeginn beim Einschalten gewährleistet.

2. Am Programmumfang sind zusätzliche Befehle nötig, die

a) den Stackpointer der CPU auf den gewünschten Anfangswert setzen (hex FF beim EMUF für die Adresse 01FF);

b) die CPU initialisieren, z. B. das Dezimal-Flag oder das Interrupt-Disable-Bit im Statusregister der CPU setzen oder rücksetzen;

c) durch das Schreiben bestimmter Werte in die Port-Datenrichtungsregister festlegen, welche Leitungen der Peripherie-Bausteine als Ein-/Ausgang dienen sollen. Am Anfang des Assemblerlistings werden ferner die nötigen Adressenvereinbarungen für den Programmspeicher (EPROM) und den Arbeitsspeicher (RAM) getroffen. Einen definierten Endpunkt weisen die Anwendungsprogramme normalerweise nicht auf, weil die Betriebssoftware ein in sich abgeschlossenes System darstellt, das nur durch Abschalten der Versorgungsspannung außer Betrieb gesetzt wird.

Gerade die folgende Applikation zeigt, daß es mit einem Mikrocomputer möglich ist, einen sonst nötigen erheblichen Schaltungsaufwand durch nur wenige ICs zu ersetzen. Im Vordergrund steht dabei die softwaremäßige Erkennung einer über Funk ausgestrahlten Tonfolge. Die Programm-Parameter sind für das europaweit zu empfangende Eurosignal-Netz ausgelegt, das auf UKW um 87,36 MHz und mit 6 stelligen Nummern arbeitet. *Abb. 7.1.1* zeigt einen Laboraufbau der Applikation, *Abb. 7.1.2.* das Flußdiagramm des EMUF-Programms.

Von der praktischen Realisation dieser Anwendung sei dem Leser allerdings abgeraten, um nicht mit den postalischen Bestimmungen in Konflikt zu geraten. Der Betrieb von Eurosignal-Empfängern ist nur mit einer Zulassung der Fernmeldebehörde gestattet und gebührenpflichtig.

Das Eurosignal-Netz dient dazu, Personen, die ziemlich viel unterwegs und daher nicht dauernd telefonisch erreichbar sind, aufzufordern, sich zu Hause, im Büro oder anderswo zu melden. Der Fernsprechteilnehmer, der einen Benutzer des europäischen

Funkrufdienstes, also den Inhaber des Empfängers erreichen will, wählt zuerst die Eurosignal-Vorwahl und dann eine 6 stellige Nummer. Diese Nummer wird dann europaweit, durch bestimmte Töne codiert, auf UKW übertragen (Abb. 7.1.3).

Die ersten fünf Ziffern der Nummer sprechen dabei das jeweilige Gerät des Teilnehmers selektiv an, während die sechste Ziffer dazu verwendet werden kann, um dem Teilnehmer klar zu machen, wo er nun anrufen soll (z. B. „1“ für zu Hause, „2“ für Büro usw.). Zu diesem Zweck leuchten unterschiedliche Leuchtdioden am Empfänger auf [17].

Insgesamt arbeitet das Eurosignal-Netz mit zwölf unterschiedlichen Tönen. Zehn davon codieren die Ziffern 0...9; einer dient als Ruhelage-Ton, solange keine Nummer übertragen wird. Ein weiterer Ton kennzeichnet die Wiederholung der vorhergehenden Ziffer: Wird beispielsweise die Nummer 122345 übertragen, so wird tatsächlich folgende Codierung ausgestrahlt: 1. 2. Wiederholung. 3, 4, 5. Jeder Ton ist dabei 100 ms lang.

Abb. 7. 1. 4 zeigt die Schaltung des Labormusters. Bei der hier vorgestellten Problemlösung sind zehn Leuchtdioden für die sechste übertragene Ziffer vorhanden, und mit zwei Drahtbrücken können vier unterschiedliche Nummernfolgen programmiert werden, die am Anfang des EPROM-Adressenbereiches gespeichert sind. Stimmen die ersten fünf Ziffern mit der Programmierung überein, so leuchtet die der sechsten Ziffer entsprechende LED auf, und etwa vier Sekunden lang ertönt ein unterbrochener Alarmton, um den Geräteinhaber auf einen empfangenen Ruf aufmerksam zu machen. Während der Lautsprecher automatisch wieder verstummt, leuchtet die LED so lange weiter, bis man die Löschtaste drückt. Treffen bis zum Drücken der Löschtaste mehrere Rufe ein, die sich in der letzten Ziffer unterscheiden, so leuchten entsprechend auch mehrere LEDs auf.

Das Programm in Abb. 7.1.5 ermöglicht einen einfachen Selbsttest, indem man während des Einschaltens die Löschtaste gedrückt hält: Dann werden nämlich probe-weise nacheinander alle LEDs kurz eingeschaltet, und auch der Alarmton ist kurz zu hören. Hält man jetzt die Löschtaste immer noch gedrückt, so ist das Gerät nicht auf die normale Rufnummer programmiert, sondern auf die Nummernfolge 12345, was durch zehnmaliges kurzes Aufblinker der Leuchtdiode „5“ angezeigt wird. Damit ist es möglich, zu überprüfen, ob die Empfangsfeldstärke für eine einwandfreie Codierung ausreicht: Im Abstand von einigen Sekunden sendet das Eurosignal-Netz nämlich als Testnummernfolge die Ziffern 123455. Wenn diese nun im Selbsttest-Modus empfangen wird, gibt der eingebaute Lautsprecher den Alarmton von sich, und die LED „5“ leuchtet als Bestätigung auf. Möchte man das Gerät nun auf die normale Rufnummer zurückprogrammieren, so schaltet man es kurz aus und wieder ein.

Sehen wir uns nun die Abschnitte des Assembler-Programms etwas näher an. Gleich am Anfang des EPROM-Speicherbereichs 0C00 stehen die vier vorprogrammierbaren Rufnummern 40673, 54333, 51401 und 47393. Jede Ziffer ist dabei in einem Byte codiert. Da jede Rufnummer nur aus fünf Ziffern besteht, für das Programm eine Adressierung aber in Achterschritten einfacher ist, wurden nach jeder Rufnummer drei Bytes mit irrelevanter Information aufgefüllt. Ab der Adresse 0C1D mit dem symbolischen Namen CHK findet sich schließlich noch die Testnummernfolge 12345.

Der Resetvektor bei 0FFC zeigt hier auf die Adresse 0C22. Das bedeutet, daß der Mikrocomputer beim Einschalten automatisch mit der Abarbeitung des ab 0C22 stehenden Programms beginnt. Er initialisiert zunächst die Ein- und Ausgabeleitungen und, soweit nötig, die CPU-Register.

Die Leitungen PB4 und PB5 werden dazu verwendet, um mit Drahtbrücken die gewünschte Rufnummer zu programmieren. Dies geschieht nach folgendem Schema (0 = Masse, 1 = offen oder + 5 V):

PB5	PB4	Nf	Vorprogrammierte Ziffernfolge
0	0	0	40673
0	1	1	54333
1	0	2	51481
1	1	3	47393

Die Rufnummer wird nun aus dem EPROM-Speicherbereich in fünf aufeinanderfolgende Zellen des Arbeitsspeichers (hier ab der Adresse 0000) übernommen. Falls aber nach dem Einschalten die Löschtaste gedrückt ist, erfolgt ein Sprung zu dem für den Selbsttest zuständigen Programmteil.

Die Decodierung der als Tonfolge empfangenen Rufnummer erfolgt nach dem Prinzip der Periodendauermessung. Dies geschieht, um eine hohe Auswertesicherheit zu erzielen, in der Routine VER sicherheitshalber zweimal nacheinander, wobei sich die beiden Meßwerte nur geringfügig voneinander unterscheiden dürfen; andernfalls wird die Messung wiederholt.

Die Zuordnung der Codierung zur Periodendauer der empfangenen Töne erfolgt über die Datentabelle FQ (0D46...0D51). Der Mikrocomputer kehrt sofort zur Hauptprogrammenschleife ML zurück, wenn die empfangene Codierung nicht mit der vorprogrammierten Rufnummer übereinstimmt. Außerdem wird, solange die Ruhelage-Frequenz empfangen wird, das Vorhandensein einer ausreichenden Empfangsfeldstärke mit einer Leuchtdiode angezeigt. Sie erlischt natürlich jedesmal kurz, wenn der Sender den Ruhelagetone unterbricht und eine Rufnummer ausstrahlt.

Rein prinzipiell wäre es natürlich auch möglich gewesen, den Alarmton für den Lautsprecher ebenfalls vom Mikrocomputer erzeugen zu lassen. Dies hätte aber Zeitprobleme verursacht: Solange der Computer einen Ton erzeugt, hat er kaum Zeit, gleichzeitig eine empfangene Tonfolge zu decodieren. Die Tonerzeugung wurde deshalb mit einer kleinen externen Schaltung vorgenommen, die ein unterbrochenes Alarmsignal an den Lautsprecher ausgibt. Sie wird im Bedarfsfall etwa vier Sekunden lang über eine Ausgangsleitung des Computers aktiviert.

Die Schaltung enthält nicht den für den UKW-Empfang zuständigen Teil. Dieser muß die empfangenen Töne mit TTL-Pegel an einen Eingang des Computers liefern, um die Periodendauer-Messung zu ermöglichen. Die Stromversorgungsschaltung ist für den Betrieb an einem 12-V-Kfz-Bordnetz ausgelegt. Sie verbraucht maximal etwa 0.4 A. Für den Betrieb im Auto spielt dies keine Rolle, für den bei tragbaren Geräten aber üblichen Batteriebetrieb wäre dies zuviel. Durch den Einsatz eines CMOS-Mikrocomputers ließe sich dieses Problem aber lösen.

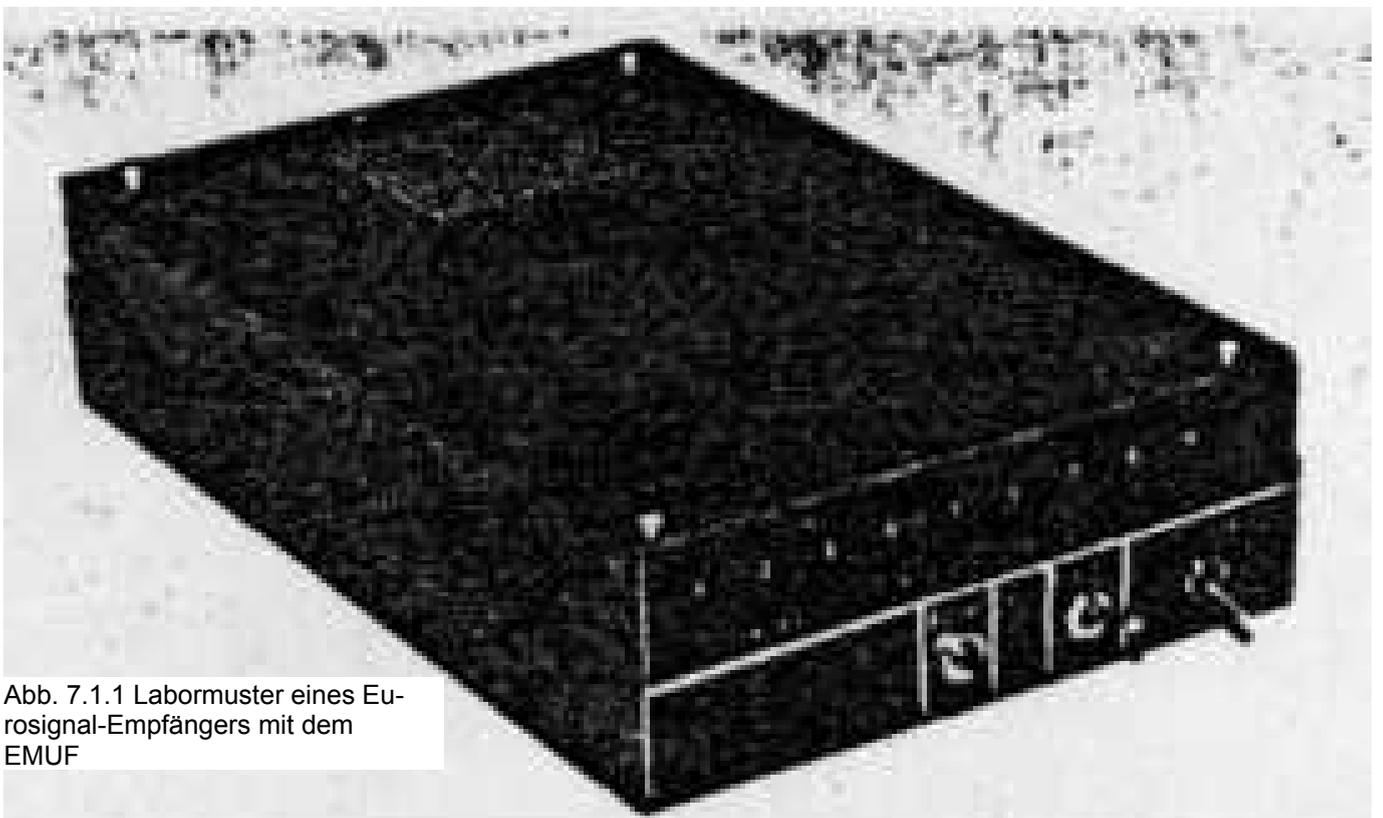


Abb. 7.1.1 Labormuster eines Euro-signal-Empfängers mit dem EMUF

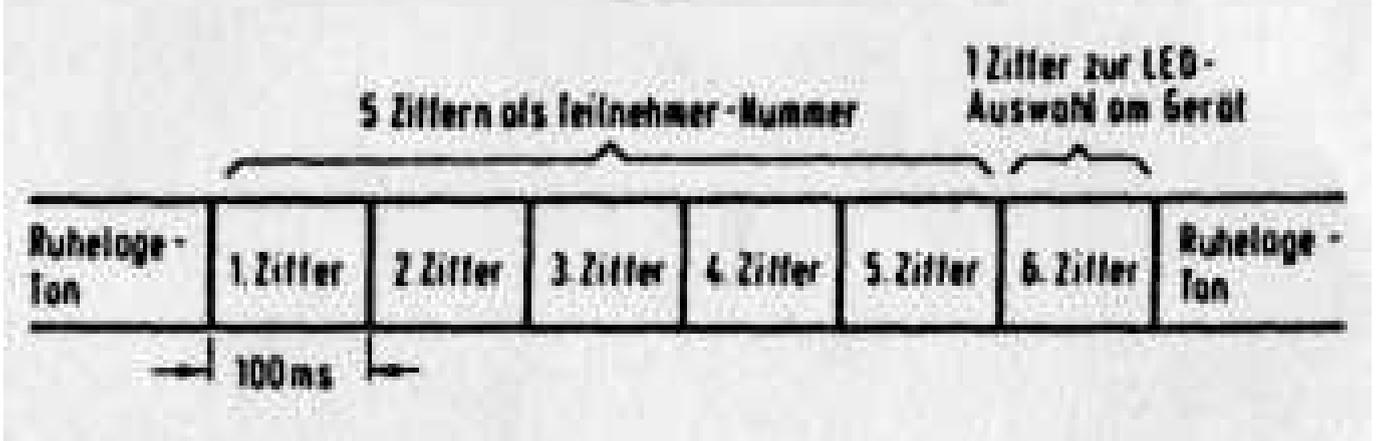


Abb. 7.1.3 Format der Ziffernübertragung beim Euro-signal-Funkrufdienst

Abb. 7.1.4 Schaltung des Tonrufempfängers

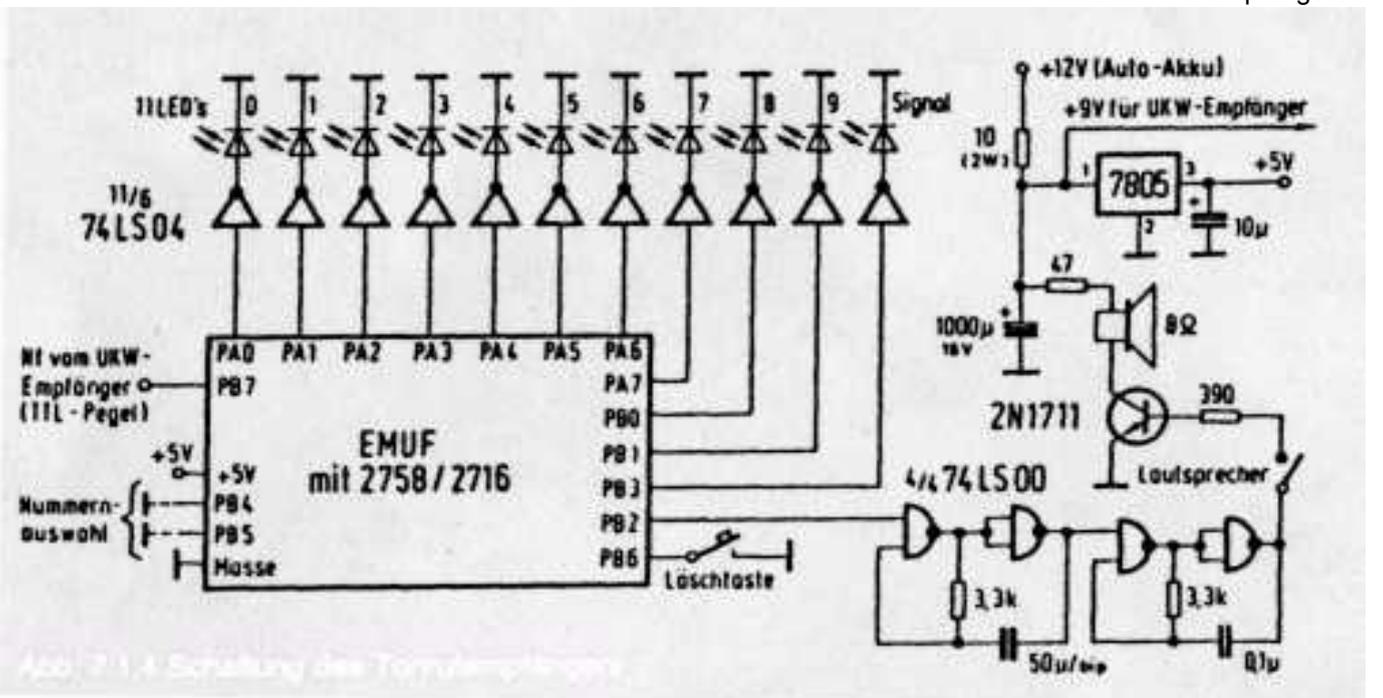


Abb. 7.1.4 Schaltung des Tonrufempfängers

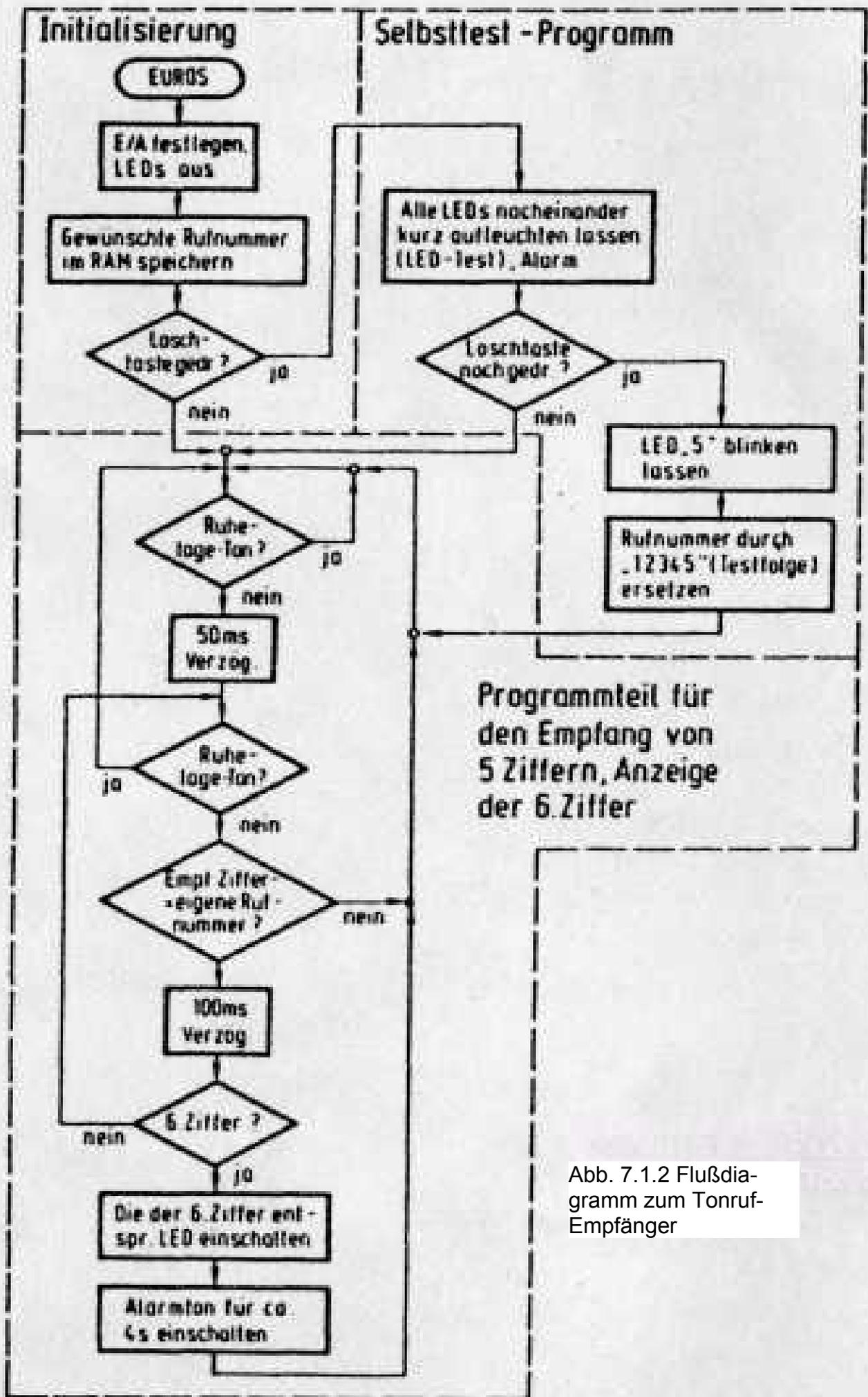


Abb. 7.1.2 Flußdiagramm zum Tonruf-Empfänger

```

FF10          PASS 1
1000          PASS 2
0000
0000          ;EUROS 1.6 - EMUF
0000 PA              =$800                      ; PORTS
0000 PAD            =$801
0000 PB              =$802
0000 PBD            =$803
0000 MSEC           =$817                      ;TIMER
0000                *=0
0000 CALL           *=*+5                      ;RAM
0005 TIL            *=*+1
0006 TIH            *=*+1
0007 FRST           *=*+1
0008 RPT            *=*+1
0009 HI              *= $C00
0C00
0C00          ;
0C00          ;EPROM
0C00 CALL0 04        .BYT 4,0,6,7,3
0C01          00
0C02          06
0C03          07
0C04          03
0C05          4E5230 .BYT 'NR0'
0C08          05        .BYT 5,4,3,3,3
0C09          04
0C0A          03
0C0B          03
0C0C          03
0C0D          4E5231 .BYT 'NR1'
0C10          05        .BYT 5,1,4,8,1
0C11          01
0C12          04
0C13          08
0C14          01
0C15          4E5232 .BYT 'NR2'
0C18          04        .BYT 4,7,3,9,3
0C19          07
0C1A          03
0C1B          09
0C1C          03
0C1D CHK 01         .BYT 1,2,3,4,5
0C1E          02
0C1F          03
0C20          04
0C21          05
0C22          ;
0C22          ;RESET SEQUENCE
0C22 RES A2FF LDX          #$FF          ;INIT
0C24          8E0108 STX          PAD          ;PORTS
0C27          9A          TXS          ;+ CPU
0C28          78          SEI
0C29          D8          CLD
0C2A          A20F LDX          #$F
0C2C          8E0308 STX          PBD          ;CLEAR
0C2F          20EE0C JSR          CLR          ;LEDS
0C32          AD0208 LDA          PB
0C35          4A          LSR          A
0C36          2918 AND          #$18
0C38          6903 ADC          #3          ;PB4-5

```

Abb. 7.1.5 Listing des Programms zur Tonfolgeruf-Decodierung

0C3D	INI	B900C	LDA	CALL0, Y	
0C40		9500	STA	CALL, X	; STORE
0C42		88	DEY		; CALL TO
0C43		CA	DEX		; Z-PAGE
0C44		10F7	BPL	INI	
0C46		2C0208	BIT	PB	; CLR KEY
0C49		7003	BVS	ML	; CLOSED?
0C4B		20090D	JSR	TEST	; YES
0C4E					
0C4E					;
0C4E					; MAIN LOOP
0C4E	ML	A200	LDX	#0	
0C50		AD0208	LDA	PB	; SWITCH
0C53		0908	ORA	#8	; MARK
0C55		8D0208	STA	PB	; LED OFF
0C58		20FA0C	JSR	NR	; MARK?
0C5B		D0F1	BNE	ML	
0C5D	ST	AD0208	LDA	PB	; MARK
0C60		29F7	AND	#\$F7	; LED
0C62		8D0208	STA	PB	; ON AGAIN
0C65		20FA0C	JSR	NR	
0C68		90E4	BCC	ML	; ERROR
0C6A		F0F1	BEQ	ST	
0C6C		A92D	LDA	#45	; HALF TONE
0C6E		8D1708	STA	MSEC	; DELAY
0C71	WAI	20400D	JSR	DLY1	
0C74	LP	A95F	LDA	#95	; ONE TONE
0C76		8D1708	STA	MSEC	; DELAY
0C79		20FA0C	JSR	NR	
0C7C		90D0	BCC	ML	; ERROR
0C7E		F0CE	BEQ	ML	; ALL OVER
0C80		88	DEY		
0C81		F002	BEQ	SM	; REPEAT
0C83		8408	STY	RPT	
0C85	SM	A408	LDY	RPT	
0C87		E005	CPX	#5	; 6TH NR?
0C89		D021	BNE	LP2	; NO
0C8B		A9FF	LDA	#\$FF	
0C8D		8509	STA	HI	
0C8F		18	CLC		; CONVERT
0C90	LP3	2A	ROL	A	; DIGIT TO
0C91		2609	ROL	HI	; BINARY
0C93		88	DEY		
0C94		D0FA	BNE	LP3	
0C96		2D0008	AND	PA	
0C99		8D0008	STA	PA	; DISPLAY
0C9C		AD0208	LDA	PB	
0C9F		2509	AND	HI	
0CA1		0904	ORA	#4	; ALARM ON
0CA3		8D0208	STA	PB	
0CA6		A90A	LDA	#10	; ALARM
0CA8		8506	STA	TIH	; CA 4SEC
0CAA		D0A2	BNE	ML	
0CAC	LP2	88	DEY		; ADJ NR
0CAD		98	TYA		; 0=0
0CAE		D500	CMP	CALL, X	
0CB0		D09C	BNE	ML	
0CB2		E8	INX		
0CB3		4C710C	JMP	WAI	; NEXT NR
0CB6					;
0CB6					; VERIFY PERIOD TIME

0CB9		8507	STA	FRST	;2PER.
0CBB		20CB0C	JSR	T	
0CBE		38	SEC		
0CBF		E507	SBC	FRST	;EQUAL?
0CC1		18	CLC		;ALLOW
0CC2		6902	ADC	#2	;-2/+1
0CC4		29FC	AND	#%11111100	
0CC6		D0EE	BNE	VER	;NO
0CC8		A507	LDA	FRST	
0CCA		60	RTS		;YES
0CCB					;
0CCB					;TEST PERIOD TIME
0CCB	T	C605	DEC	TIL	;ALARM
0CCD		D00C	BNE	T0	;TIME
0CCF		C606	DEC	TIH	;OVER?
0CD1		D008	BNE	T0	
0CD3		AD0208	LDA	PB	
0CD6		29FB	AND	#\$FB	;ALARM OFF
0CD8		8D0208	STA	PB	
0CDB	T0	20E00C	JSR	T1	;SYNC
0CDE		A000	LDY	#0	
0CE0	T1	C8	INY		;MEASURE
0CE1		2C0208	BIT	PB	;1ST HALF
0CE4		30FA	BMI	T1	;CYCLE
0CE6	T2	C8	INY		;AND NOW
0CE7		2C0208	BIT	PB	;2ND
0CEA		10FA	BPL	T2	
0CEC		700A	BVS	T3	;CLR LEDS?
0CEE	CLR	A9FF	LDA	#\$FF	;YES
0CF0		8D0008	STA	PA	
0CF3		A90B	LDA	#\$B	;ALARM OFF
0CF5		8D0208	STA	PB	
0CF8	T3	98	TYA		
0CF9		60	RTS		;T IN Y+A
0CFA					;
0CFA					;DETERMINE NUMBER
0CFA	NR	20B60C	JSR	VER	
0CFD		A00B	LDY	#11	
0CFF	T4	D9460D	CMP	FQ,Y	
0D02		B003	BCS	RTN	
0D04		88	DEY		
0D05		10F8	BPL	T4	;C=0:ERROR
0D07	RTN	98	TYA		;A=Y=NR
0D08		60	RTS		;Z=1 :MARK
0D09					;
0D09					;SELF-TEST
0D09	TEST	A20B	LDX	#11	;TEST
0D0B		18	CLC		;ALL NR-
0D0C		9004	BCC	TST0	;LEDS
0D0E	TST1	203B0D	JSR	DLY	
0D11		38	SEC		;AND
0D12	TST0	2E0008	ROL	PA	;ALARM
0D15		2E0208	ROL	PB	;ON/OFF
0D18		CA	DEX		
0D19		D0F3	BNE	TST1	
0D1B		2C0208	BIT	PB	;CLR KEY
0D1E		70E7	BVS	RTN	;DOWN?
0D20		A214	LDX	#20	;YES
0D22	TST2	AD0008	LDA	PA	
0D25		4920	EOR	#\$20	;FLASH
0D27		8D0008	STA	PA	;NR.5

```

0D2D      CA      DEX                      ;10TIMES
0D2E      D0F2   BNE                      TST2
0D30      A204   LDX                      #4
0D32  TST3  BD1D0C LDA                      CHK,X
0D35      9500   STA                      CALL,X ;USE TEST
0D37      CA      DEX                      ;SEQUENCE
0D38      10F8   BPL                      TST3 ;AS CALL
0D3A      60      RTS
0D3B  DLY   A062   LDY                      #98 ;DELAY
0D3D      8C1708 STY                      MSEC ;0.1 SEC
0D40  DLY1  2C1708 BIT                      MSEC
0D43      10FB   BPL                      DLY1 ;WAIT
0D45      60      RTS
0D46      ;
0D46      ;FREQ VALUES
0D46  FQ    5A      .BYT 90,99,108
0D47      63
0D48      6C
0D49      74      .BYT 116,128,139,151
0D4A      80
0D4B      8B
0D4C      97
0D4D      A4      .BYT 164,178,193
0D4E      B2
0D4F      C1
0D50      D1      .BYT 209,227
0D51      E3
0D52      FF      .BYT $FF,$FF,$FF
0D53      FF
0D54      FF
0D55      4443   .BYT 'DC1YB'
0D5B      4555   .BYT 'EUROS 1.6'
0D64      FF      .BYT $FF,$FF,$FF
0D65      FF
0D66      FF
0D67      ;
0D67      •=$FFC
0FFC
0FFC      ;RESET VECTOR
0FFC      220C   .WOR RES
0FFE      220C   .WOR RES
1000      .END

```