Assembler ist keine Alchimie — Teil 12

Im Gegensatz zum sonstigen Sprachgebrauch erregt das Kürzel CIA bei Commodore 64-Kennern angenehme Assoziationen. Die beiden CIAs (Complex Interface Adapter) unseres Computers und ihre Rolle bei der Unterbrechungs-Programmierung sollen in dieser Folge entschleiert werden.

Lassen Sie uns kurz rekapitulieren: Als primäre Unterbrechungsanforderer hatten wir drei Bausteine unseres Computers benannt, nämlich den VIC-II-Chip und die beiden CIA-Bausteine. CIA kommt von »Complex Interface Adapter« und ist die Bezeichnung für die beiden Ein- und Ausgabe-Bausteine, die den gesamten Verkehr zwischen dem zentralen Gehirn unseres C 64 und der Peripherie managen. Wir hatten bemerkt, daß ein CIA, der IRQ-CIA (Adressen von 56320 bis 56575), ausschließlich für die maskierbaren Unterbrechungen zuständig ist. Dazu gehören die 60mal pro Sekunde stattfindenden »Timer-Interrupts«, die die Cursorbehandlung, die TI$-Uhr, die Tastaturabfrage etc. bearbeiten. Der andere CIA, genannt NMI-CIA, (Adressenraum 56576-56831) ist nur für die nicht maskierbaren Unterbrechungen verantwortlich und wird bei normaler Nutzung des C 64 so gut wie nie eingesetzt. Ich gehe im folgenden davon aus, daß Sie keine RS232C-Schnittstelle in Ihren Computer eingesetzt haben. Sollte das aber der Fall sein, dann müßten Sie darauf achten, die folgenden Beispiele — die den NMI-CIA betreffen — ohne gleichzeitigen Betrieb dieser Schnittstelle anzuwenden, weil sich sonst Störungen ergeben könnten.

In der Folge 10 dieser Serie (64'er, Ausgabe 7/85) haben wir uns ein Register (das Register 13, Interrupt-Kontrollregister) der CIAs schon genauer angesehen und auch die Unterschiede beider Bausteine festgestellt. Dort war dann die Rede von Timern, Echtzeituhren, Alarm-Funktionen etc. Was es damit auf sich hat und wie man diese Möglichkeiten nutzen kann, das soll diesmal unser Thema sein. Wir werden uns dazu alle Register der CIAs genauer ansehen, die für die von uns ausgewählten Unterbrechungsoptionen eine Rolle spielen. Dabei fallen einige unter den Tisch — das habe ich aber schon in Folge 10 angekündigt —, nämlich diejenigen, die mit dem Verkehr über den seriellen Port, beziehungsweise über die RS232C-Schnittstelle, zu tun haben. Es bleibt dann anderen — kompetenteren — überlassen, darüber zu schreiben.

Register Nr. ($)	Adresse (dez.) CIA-1	CIA-2	Name	Funktion
04	56324	56580	TALO	TIMER A LSB
05	56325	56581	TAHI	TIMER A MSB
06	56326	56582	TBLO	TIMER B LSB
07	56327	56583	TBHI	TIMER B MSB
08	56328	56584	TOD10THS	1/10-Sekunden-Register
09	56329	56585	TODSEC	Sekunden-Register
0A	56330	56586	TODMIN	Minuten-Register
0B	56331	56587	TODHR	Stunden-Register, AM/PM-Flagge
0D	56333	56589	JCR	Unterbrechungs-Kontrollregister
0E	56334	56590	CRA	Kontrollregister A
0F	56335	56591	CRB	Kontrollregister B

Tabelle 1. Die wichtigen Register der beiden CIAs

Wie wäre es zum Beispiel mit Ihnen?

Auch so bleibt uns genug zu tun. In Tabelle 1 finden Sie zunächst eine Übersicht der von uns behandelten Register.

Sie sehen darin, daß jeder CIA über zwei sogenannte Timer (A und B) verfügt, sodann über die »Time of Day« (zu deutsch etwa »Tageszeit«) genannte Echtzeituhr mit vier Registern und schließlich noch über drei Kontrollregister, zu denen auch das schon erwähnte Register 13 gehört. Sehen wir uns zunächst die Timer an.

Die Timer der CIAs.

Insgesamt verfügen wir über vier dieser Timer: Timer A und B im CIA1 und dasselbe nochmal im CIA2. Es handelt sich dabei um 16-Bit-Register, in die ein Startwert geschrieben werden kann, von dem an dann heruntergezählt wird. Jedesmal, wenn dann der Wert 0 unterschritten ist, gibt es für uns die Möglichkeit, bestimmte Ereignisse stattfinden zu lassen. Man kann diese Register unabhängig voneinander, aber auch kombiniert, benutzen. Ein Lesen des Registers liefert immer den momentan gerade aktuellen Wert. Ein Schreiben in das Register führt automatisch zum Festlegen eines Startwertes. Was an Optionen mit diesen Timern möglich ist, wird über Kontrollregister gesteuert. Das CRA (Register $ 0E) bezieht sich vor allem auf den Timer A, das CRB (Register $ 0F) auf Timer B. Die 16-Bit-Register werden — wie gewohnt — in der Form LSB/MSB betrieben. In den Timer A des CIA1 wird bei jedem I/O-Reset folgendes Wertepaar eingetragen:

56324 dezimal 37 LSB
56325 dezimal 64 MSB

Das entspricht einem Startwert von 16421. Im PAL-System hat der Quarz, der die Taktfrequenz bestimmt, eine Frequenz von 17.734472 MHz. Die Prozessorfrequenz errechnet sich daraus mittels Division durch 18 zu 985248.4 Hz (also etwas weniger als 1 MHz, was den europäischen C 64 langsamer macht als den amerikanischen, der etwas mehr als 1 MHz verwendet). Wenn mit dieser Geschwindigkeit der Timer heruntergezählt wird, erhält man genau einen Unterlauf alle 1/60 Sekunden. Das ist der Weg, eine kontrollierte Zeitspanne durch den Timer zählen zu lassen. Sei X der gesuchte Startwert, der zu einer Spanne von T Sekunden führt, dann kann man X berechnen mittels:

X = 985248.4 * T

Der Integerwert von X ist dann in ein LSB und ein MSB zu teilen und in die Timer-Register einzutragen. Allerdings ergibt sich so eine natürliche Grenze. Die höchste durch 2 Bytes darstellbare Zahl ist ja 65535. Wenn wir diesen Wert in den Timer schreiben, dann ist er alle 1/15 Sekunden auf 0 heruntergezählt. Für längere Zeiten ist aber vorgesorgt. Die beiden Timer A und B sind kombinierbar (wie, dazu kommen wir gleich noch) zu einem 32-Bit-Register. Die höchste Zahl X ist dann:

4 294 967 296 = 2^32

Damit kann im Extremfall eine Herabzählzeit von 1 Stunde, 12 Minuten und zirka 40 Sekunden eingeplant werden, was für die meisten Zwecke ausreichen dürfte.

Möchten Sie also genau eine Sekunde Spielraum haben beim Herunterzählen, dann muß die Zahl 985248 als 4-Byte-Integer-Wert in die Speicher von Timer A und Timer B gebracht werden. Das führt dann zu den Werten 0, 15, 8, 160 (weil 985248 = 0*16777216 + 15*65536 + 8*256 + 160). 0 und 15 gelangen als MSB beziehungsweise LSB in Timer B (also Register 07 und 06), 8 und 160 sind MSB und LSB für den Timer A (Register 05 und 04). Sehen wir uns nun an, wie wir dem Computer sagen, was mit diesen Startwerten in den Timer-Registern geschehen soll. Die beiden Kontrollregister CRA und CRB beziehen sich weitgehend auf die gleichnamigen Timer. Im Bild 1 finden Sie das Register $0E, also CRA und in Bild 2 das andere Kontrollregister CRB ($0F):

Die Bedeutung der Bits 0 bis 4 ist — jeweils für den dazugehörigen Timer — identisch:

Bit 0
an dieser Stelle führt zum sofortigen Anhalten des Timers. 1 in diesem Bit startet das Herunterzählen.

Bits 1 und 2
Diese beiden Bits hängen mit dem externen Signalverkehr zusammen und sollen für uns außer acht bleiben.

Bit 3
Ist dieses Bit = 1, dann ist der sogenannte »One Shot«-Betrieb des Timers aktiv. Das bedeutet, daß vom Startwert an heruntergezählt wird bis auf Null. Es findet nun das programmierte Ereignis statt (zum Beispiel ein IRQ). Anschließend wird der Startwert wieder eingeladen und der Timer gestoppt.

Im Gegensatz dazu läuft der »Continuous«-Betrieb, wenn das Bit den Wert 0 enthält. Dabei geschieht zunächst dasselbe wie beim One Shot Modus, der Timer wird aber nicht angehalten, sondern der ganze Vorgang wiederholt sich in einer Endlosschleife.

Bit 4
Ein Hineinschreiben einer 1 in dieses Bit erzeugt ein sofortiges Neuladen der Timer-Register mit dem Startwert. Dabei ist es gleichgültig, ob der Timer gerade läuft oder nicht. Schreibt man eine Null ein, hat das keine Wirkung.

Beim Lesen des Registers ist dieses Bit immer 0.

Zu diesem Bit und seiner Wirkung ist noch etwas zu sagen. Das Neuladen des Timers geschieht
— immer dann, wenn ein Unterlauf stattgefunden hat oder
— falls der Timer steht und in die Register ein Startwert geschrieben wird. Dabei ist der CIA so konstruiert, daß man kein zwangsweises Laden (also mit Bit 4 = 1) braucht, wenn man den Startwert in der Reihenfolge LSB MSB in die Register bringt.

Die Bits 5 bis 7 haben nun unterschiedliche Bedeutung im CRA und im CRB:

Register CRA ($0E)

Bit 5:

Ist dieses Bit gleich Null, dann wird im Systemtakt gezählt. Den hatten wir vorhin zur Zeitberechnung schon verwendet. Wenn das Bit auf 1 gesetzt ist, zählt der Timer externe Signale.

Bit 6:

Spielt für den Signalverkehr über den seriellen Port eine Rolle und soll uns hier nicht weiter beschäftigen.

Bit 7:

Damit steuert man nicht den Timer A, sondern dieses Bit bezieht sich auf die gleich noch zu behandelnde Echtzeituhr.

Register CRB ($0F)

Die Bits 5 und 6 sind hier im Zusammenhang von Bedeutung. Es gibt vier Kombinationsmöglichkeiten:

Bit 6 — Bit 5

0 — 0

Der Timer B wird — wie vorhin der Timer A — im Systemtakt heruntergezählt.

0 — 1

Der Timer B wird durch externe Signale heruntergezählt.

1 — 0

Der Timer B zählt die Unterläufe von Timer A. Das ist der vorhin erwähnte Punkt, der beide Timer kombiniert zum 32-Bit-Zähler. Man kann also im Extremfall 65536 mal 65536 Takte zählen lassen.

1 — 1

Auch in diesem Fall zählt Timer B die Unterläufe von Timer A. Er tut das aber nur, wenn ein bestimmtes externes Signal vorhanden ist.

Bit 7:

Auch beim Register CRB steuert dieses Bit bestimmte Möglichkeiten der Echtzeituhr. Deshalb haben Sie noch ein wenig Geduld, bis wir diese Uhr behandeln.

Wir kennen uns nun ganz gut aus, wie wir mit den Timern umzugehen haben. Unser Wissen soll in einem kleinen Test erprobt werden. Dazu bedienen wir uns des ‰ Sekunden IRQ. Wir verändern diese regelmäßige Unterbrechung derart, daß sie nur noch einmal in der Sekunde geschieht. Welche Zahlen dazu in ein 32-Bit-Register gepackt werden müssen, haben wir schon vorhin berechnet. Jeweils in der Reihenfolge LSB/MSB müssen wir sie einschreiben und vorher die Timer anhalten, indem die Bits 0 der Kontrollregister CRA und CRB auf 0 gesetzt werden. Nach dem Einschreiben und Starten der beiden Timer müssen folgende Bitmuster in CRA und CRB stehen:

CRA
Bit 0 = 1 Start Timer A
Bit 3 = 0 Dauerlauf
Bit 5 = 0 Systemtakt

CRB
Bit 0 = 1 Start Timer B
Bit 3 = 0 Dauerlauf
Bit 5 = 0
Bit 6 = 1 Timer B zählt Unterläufe von Timer A.

Bevor wir die Timer starten, muß auch noch das Interrupt-Kontrollregister verändert werden (das hatten wir uns in der 10. Folge genauer angesehen). Bislang erzeugt ein Unterlauf des Timer A eine Unterbrechung. Wir möchten aber, daß der Timer B (damit wir das 32-Bit-Register voll ausnutzen) der Auslöser ist. Dazu muß Bit 0 des ICR gelöscht und statt dessen Bit 1 gesetzt werden.

Im Programm »Timer-Test« (siehe Listing 1 und 2) ist all das realisiert. Mit SYS 49152 gestartet, zeigt sich sofort ein deutlich verlangsamter Cursor. Noch langsamer kann alles werden, indem Sie höhere Werte in die Timer-Register schreiben. Den Normalzustand stellen Sie einfach durch Drücken der RUN/STOP- und der RESTORE-Tasten her. Dabei wird ja — wie Sie aus der letzten Folge her wissen, auch ein I/O-Reset ausgeführt, der den Ausgangszustand wiederherstellt.

Die Verlängerung des IRQ-Zyklus hat übrigens noch einen sinnvollen Nebeneffekt. Je seltener ein laufendes Programm unterbrochen wird, desto schneller wird es mit seinen Jobs fertig. Das kann man immer dann tun — im Extremfall sogar den IRQ ganz ausschalten — wenn man die Möglichkeiten, die der Computer während des normalen IRQ anbietet, nur selten oder aber gar nicht braucht.

Die Echtzeituhren

Wir kennen nun fünf Uhren in unserem Computer: Die vier Timer (jeweils A und B im CIA1 und CIA2), die wir, weil wir die Impulszahlen in Zeiteinheiten umrechnen können, zur Zeitmessung einsetzen könnten und die im Basic verfügbare Uhr TI$, die aber — wie wir nun wissen — lediglich die Umsetzung des Timer A im CIA1 in ein bequemer handhabbares Software-Instrument ist. Zudem ist die Ganggenauigkeit dieser Uhr recht gering. Schon einige Kassettenoperationen genügen, sie völlig aus dem Takt zu bringen.

Um so mehr verwundert es, daß zwei hervorragende Echtzeituhren im Commodore 64 so gut wie nie benutzt werden, ja nicht einmal in irgendeiner Weise softwaremäßig unterstützt werden. Vielleicht ist das ein bißchen zuviel »mehr sein als scheinen«, was Commodore da betreibt, wenn man bedenkt, welche verborgenen Schätze da alle zutage gefördert werden können (man denke nur an die hochauflösende Grafik) bei genauer Untersuchung des Computers.

Jeder der beiden CIAs verfügt über solch eine Uhr, die direkt von der Netzfrequenz getaktet wird. Die Zählung der Zeit geschieht in vier Registern (Register $08 bis $0B), die in Bild 3 gezeigt sind.

Vielleicht fällt Ihnen etwas auf, wenn Sie sich diese vier Bytes mal genauer ansehen: Die Speicherung geschieht in Form von Einer- und Zehnerstellen. Das kann also weder im Binärformat noch als ASCII-Zeichen funktionieren. Hier werden die Ziffern als BCD-Zahlen abgelegt. In der 3.Folge dieser Serie wurde dieses »binary coded decimal«-Format erklärt. Das ist lange her (64'er, Ausgabe 11/84) und soll deshalb hier nochmal vorgestellt werden, damit alle wissen, wovon die Rede ist.

In dieser Zahlendarstellung wird jede Dezimalstelle einer Zahl gesondert in eine Binärzahl umgewandelt. Dann ergibt sich der folgende Zusammenhang:

Binär	Dezimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

Das war's! Die anderen möglichen Binärkombinationen (also zum Beispiel 1010 etc.) werden nicht benutzt. Die Zahl 25 beispielsweise lautet im BCD-Format:

0010 0101
^ ^
2 5

Jetzt ist es Ihnen sicherlich verständlich, warum für die Sekunden- und Minuten-Zehnerstellen nicht mehr als drei Bits reserviert wurden: größer als 6 wird die Zehnerstelle nicht.

Zum Stundenregister TODHR ist aber noch etwas zu sagen: Dort ist nur ein Bit reserviert für die Stunden-Zehnerstelle. Die Uhr läuft nicht bis 24 Uhr, sondern lediglich bis 12 Uhr. Zur Unterscheidung, ob vor- oder nachmittags gemeint ist, dient das Bit 7. Dieses sogenannte AM/PM-Flag ist orientiert an der angelsächsischen Gewohnheit, zum Beispiel für 16 Uhr den Ausdruck 4 PM zu verwenden. PM kommt vom lateinischen »post meridiem«, was übersetzt heißt »nach dem Mittag«, wohingegen AM steht für »ante meridiem«, also »vor dem Mittag«. Meint man nun AM, dann muß diese Flagge auf 0, bei PM aber auf 1 gesetzt sein.

Beim Stellen der Uhren sollte eine Reihenfolge eingehalten werden. Sobald nämlich in das Stundenregister geschrieben wird, hält die Zählung automatisch an. Man kann nun die anderen Werte in die Register schreiben. Den Startschuß liefert das Schreiben in das Register

```
programm : prg.timer-testc000 c05i
-----------------------------------

c000 : 78 ad 0e dc 29 fe 8d 0e  4b
c008 : dc ad 0f dc 29 fe 8d 0f  f9
c010 : dc a9 0f 8d 06 dc a9 00  24
c018 : 8d 07 dc a9 a0 8d 04 dc  d5
c020 : a9 08 8d 05 dc a9 1f 8d  84
c028 : 0d dc a9 82 8d 0d dc ad  6e
c030 : 0e dc 29 d7 8d 0e dc ad  0a
c038 : 0f dc 29 d7 8d 0f dc ad  1b
c040 : 0e dc 09 01 8d 0e dc ad  37
c048 : 0f dc 09 41 8d 0f dc 58  a5
c050 : 60 ff 00 ff 00 ff 00 ff  b0
```

Listing 1. Programm Timer-Test, ein Beispiel für die Anwendung eines 32-Bit-Timers

```
programm : obj.alarmuhr c000 c18d
----------------------------------

c000 : a9 8e 8d 11 03 a9 c0 8d  11
c008 : 12 03 a9 1d 8d 18 03 a9  a3
c010 : c0 8d 19 03 ad 0e dd 09  12
c018 : 80 8d 0e dd 60 48 8a 48  a1
c020 : 98 48 a9 7f 8d 0d dd ac  49
c028 : 0d dd 10 06 4c 6a c1 4c  a0
c030 : 72 fe 20 bc f6 20 e1 ff  b9
c038 : d0 f5 a2 04 bd 2f fd 9d  b4
c040 : 13 03 ca d0 f7 a2 1a bd  1a
c048 : 35 fd 9d 19 03 ca d0 f7  c0
c050 : a9 7f 8d 0d dc 8d 0d dd  e8
c058 : 8d 00 dc a9 08 8d 0e dc  30
c060 : a9 88 8d 0e dd a9 08 20  fe
c068 : b6 fd 4c 6c fe a9 48 8d  37
c070 : 11 03 a9 b2 8d 12 03 78  2a
c078 : a9 47 8d 18 03 a9 fe 8d  c0
c080 : 19 03 a9 31 8d 14 03 a9  84
c088 : ea 8d 15 03 58 60 24 0d  12
c090 : 30 03 4c 20 c1 20 82 b7  f0
c098 : c0 07 d0 40 ad 0f dd 29  35
c0a0 : 7f 8d 0f dd a0 00 a9 24  5e
c0a8 : 20 fe c0 d0 02 a9 24 c9  23
c0b0 : 13 90 07 f8 38 e9 12 d8  b9
c0b8 : 09 80 8d 0b dd 20 fc c0  1a
c0c0 : 8d 0a dd 20 fc c0 8d 09  ec
c0c8 : dd 20 66 c1 8d 08 dd a9  6b
c0d0 : 00 4c 3c bc 68 68 68 68  d9
c0d8 : a9 ff d0 f5 c0 08 d0 f8  f5
c0e0 : ad 0f dd 09 80 8d 0f dd  1a
c0e8 : a9 84 8d 0d dd a9 3c 85  ff
c0f0 : 04 85 02 a9 ff 85 03 a0  e6
c0f8 : 01 4c a6 c0 a9 60 85 24  dd
c100 : 20 13 c1 0a 0a 0a 0a 85  80
c108 : 25 20 13 c1 05 25 c5 24  13
c110 : b0 c4 60 b1 22 38 e9 30  5d
c118 : 90 ba c9 0a b0 b6 c8 60  5e
c120 : a9 07 20 7d b4 a0 00 ad  b0
c128 : 0b dd 08 29 1f c9 12 d0  73
c130 : 02 a9 00 28 10 05 f8 18  49
c138 : 69 12 d8 20 55 c1 ad 0a  13
c140 : dd 20 55 c1 ad 09 dd 20  96
c148 : 55 c1 ad 08 dd 20 60 c1  ce
c150 : 68 68 4c ca b4 48 4a 4a  a4
c158 : 4a 4a 20 60 c1 68 29 0f  fe
c160 : 09 30 91 62 c8 60 20 13  68
c168 : c1 60 a9 77 8d 14 03 a9  8b
c170 : c1 8d 15 03 4c bc fe c6  d2
c178 : 02 f0 03 4c 31 ea a5 04  46
c180 : 85 02 ad 20 d0 45 03 8d  d4
c188 : 20 d0 4c 31 ea 00 ff 00  f8
```

Listing 3. Eine Echtzeituhr. Bitte beachten Sie die Eingabehinweise auf Seite 54

Listing 2. Der Quelltext zum Timer-Set.

```
PASS 1

PASS 2
7000           0823 ;
7000           084C ;***************************************
7000           0375 ;*                                     *
7000           089E ;*              TIMER-TEST             *
7000           08C7 ;*                                     *
7000           08F0 ;* TIMER A UND B DES CIA1 WERDEN SO    *
7000           0919 ;* GESCHALTET, DASS NUR NOCH 1 MAL     *
7000           0942 ;* PRO SEKUNDE DER TIMER-IRQ AUFTRITT  *
7000           096B ;*                                     *
7000           0994 ;*    HEIMO PONNATH  HAMBURG  1985     *
7000           09BD ;*                                     *
7000           09E6 ;***************************************
7000           09E9 ;
7000           09EC ;
C000           03F8                .BA $C000
C000           09FE                .OS
C000           0A01 ;
C000           0A27 ;+++ BENUTZTE ADRESSEN DES CIA 1 +++
C000           0A2A ;
C000           0A3A TALO           .DE $DC04
C000           0A4A TAHI           .DE $DC05
C000           0A5A TBLO           .DE $DC06
C000           0A6A TBHI           .DE $DC07
C000           0A79 ICR            .DE $DC0D
C000           0A88 CRA            .DE $DC0E
C000           0A97 CRB            .DE $DC0F
C000           0A9A ;
C000           0AC3 ;+++ EINSCHALTEN DES 1 SEKUNDEN IRQ +++
C000           0AC6 ;
C000 78        0AE5 START          SEI                     ;SPERREN ALLER IRQS
C001           0AE8 ;
C001 AD 0E DC  0AF2                LDA CRA
C004 29 FE     0B03                AND #%11111110
C006 8D 0E DC  0B18                STA CRA                 ;STOP TIMER A
C009 AD 0F DC  0B25                LDA CRB
C00C 29 FE     0B36                AND #%11111110
C00E 8D 0F DC  0B4E                STA CRB                 ;STOP TIMER B
C011           0B51 ;
C011 A9 0F     0B6F                LDA #15                 ;NEUER STARTWERT IN
C013 8D 06 DC  0B8A                STA TBLO                ;32-BIT-REGISTER
C016 A9 00     0B94                LDA #00
C018 8D 07 DC  0B9F                STA TBHI
C01B A9 A0     0BAA                LDA #160
C01D 8D 04 DC  0BB5                STA TALO
C020 A9 08     0BBF                LDA #08
C022 8D 05 DC  0BCA                STA TAHI
C025           0BCD ;
C025 A9 1F     0BDE                LDA #%00011111
C027 8D 0D DC  0BF8                STA ICR                 ;ALLE IRQ VERBOTEN
C02A A9 82     0C0C                LDA #%10000010
C02C 8D 0D DC  0C27                STA ICR                 ;NUR TIMER B IRQ
C02F           0C2A ;
C02F AD 0E DC  0C34                LDA CRA
C032 29 D7     0C45                AND #%11010111
C034 8D 0E DC  0C61                STA CRA                 ;BITS 3 UND 5 = 0
C037           0C64 ;
C037 AD 0F DC  0C6E                LDA CRB
C03A 29 D7     0C7F                AND #%11010111
C03C 8D 0F DC  0C8F                STA CRB                 ;DITO
C03F           0C92 ;
C03F AD 0E DC  0C9C                LDA CRA
C042 09 01     0CAD                ORA #%00000001
C044 8D 0E DC  0CC6                STA CRA                 ;TIMER A START
C047           0CC9 ;
C047 AD 0F DC  0CD3                LDA CRB
C04A 09 41     0CE4                ORA #%01000001
C04C 8D 0F DC  0D01                STA CRB                 ;TIMER B START MIT
C04F           0D24 ;                                      TIMER A UNTERLAUF
C04F           0D27 ;
C04F 58        0D3D                CLI                     ;IRQS FREIGEBEN
C050           0D40 ;
C050 60        0D46                RTS
C051           0D49 ;
C051           0D4F                .EN
```

7	6	5	4	3	2	1	0
TODIN 50Hz/60 Hz	externer Signal-verkehr	in Mode	Force-load	ONE Shot / Continu-ous	externer Signal-verkehr		Start/Stop

Bild 1. Das Kontrollregister des Timer A

7	6	5	4	3	2	1	0
ALARM	In MODE		Force-load	ONE Shot / Continu-ous	externer Signal-verkehr		Start/Stop

Bild 2. Dasselbe für den Timer B

Register Name	Nr.	7	6	5	4	3	2	1	0
TOD10THS	08	unbenutzt				⅒-Sekundenwert			
TODSEC	09	unbenutzt	Zehnerstelle Sekunden			Einerstelle Sekunden			
TODMIN	0A	unbenutzt	Zehnerstelle Minuten			Einerstelle Minuten			
TODHR	0B	AM/PM Flagge	unbenutzt		Zehnerstel-le Stunden	Einerstelle Stunden			

Bild 3. Die Register der Echtzeituhren

TOD10TH: von nun an tickt die Uhr wieder.

Ähnlich funktioniert das Lesen der Uhrzeit. Sobald das Stundenregister gelesen wird, führt das zum Anhalten der Uhr, so daß die restlichen Register reibungslos auslesbar sind. Wieder ist es das Zehntelsekundenregister, das beim Auslesen ein Weiterlaufen der Uhr bewirkt. Aber, so werden Sie bemerken, wenn der Auslesevorgang eine bestimmte Zeit beansprucht, führt das zu Verzögerungen? Die Lösung ist, daß der gesamte Inhalt der vier Register gleichzeitig mit dem Auslesen des Stundenwertes in einen internen Speicher transferiert wird und dort weiterläuft. Nach dem Lesen des TOD10TH kommt der aktuelle Wert zurück in die Register und dieser wird weitergezählt.

Nun wird es höchste Zeit, daß wir uns die beiden Bits im CRA und im CRB ansehen, die wir vorhin bei der Timer-Behandlung links liegen ließen. Bit 7 im CRA kündigt der Echtzeituhr an, welche Netzfrequenz zu erwarten ist. Eine 1 an dieser Stelle steht für 50 Hz, eine 0 für 60 Hz. Unser Stromnetz in Deutschland liefert einen Wechselstrom mit 50 Hz, weshalb wir dann dort die 1 setzen sollten. Da gibt es ein kleines Problem: Beim I/O-Reset, der durch Drücken der RUN/STOP- und der RESTORE-Tasten zusammen ausgelöst wird, schreibt der Computer immer den amerikanischen Wert für 60 Hz in dieses Bit. Dann geht die Uhr aber empfindlich nach. Man muß also einen Weg finden, der erlaubt, dort in diesem Fall wieder eine 1 einzuschreiben. Das ist durch eine eigene NMI-Routine möglich. Sie sehen schon, der Weg zur Nutzung dieser verlockenden Uhren ist ziemlich dornenreich!

Noch interessanter ist das Bit 7 im CRB. Das Setzen der Uhrzeit ist nämlich nur möglich, wenn dieses Bit den Inhalt 0 hat. Was geschieht, wenn dort eine 1 steht? Dann bestimmt man nicht die aktuelle Uhrzeit, sondern man stellt einen Wecker (das ist die Alarmzeit). Das geschieht nach dem Setzen dieses Bits genauso wie vorhin das Einschreiben der Uhrzeit (also erstaunlicherweise auch in genau dieselben Register!). Im Unterschied dazu ist allerdings ein Lesen der Alarmzeit nicht möglich — das ergibt immer die aktuelle Uhrzeit. Man muß für diesen Fall die Weckzeit irgendwo abspeichern und bei Bedarf dann von dort lesen.

Weil man ja meistens nach dem Erreichen der Alarmzeit irgendeine Reaktion erwartet, ist im ICR (also dem Unterbrechungskontrollregister 13) jedes CIA noch ein Bit reserviert — das Bit 2 —, mit dessen Hilfe der Alarm per IRQ oder NMI wie auch immer geartet losbrechen kann. Der Phantasie sind hier nur wenige Grenzen gesetzt. Wie man mit diesem ICR umgeht, ist Ihnen noch aus der Folge 10 geläufig.

Damit sind wir durch die Eigenheiten der CIAs durch. Man braucht tatsächlich keine Scheu zu haben, diese Echtzeituhren zu nutzen. Lediglich die Uhr im CIA1 wird manchmal verwendet, einen bestimmten Wert für die Zufallszahlenerzeugung zu generieren. Aber das sollte einer eigenen Uhren-Routine nicht in die Quere kommen. Solch eine Echtzeituhr finden Sie im beiliegenden Listing 3 und 4.

Durch SYS49152 aktivieren Sie die Uhr, die Sie mit SYS49261 auch wieder abschalten können. Durch ein USR-Kommando A = USR (String) stellen Sie die Startzeit ein. String kann dabei eine Stringvariable sein oder auch direkt ein String der Form »HHMMSST« (also Stunden, Minuten, Sekunden, Zehntelsekunden). In A steht eine 0, wenn kein Fehler, aber eine -1, wenn ein Fehler aufgetreten ist. Das Lesen der Uhr erfolgt über ein zweites USR-Kommando: PRINTUSR(Zahl). Dabei kann Zahl eine beliebige Zahl oder Variable sein. Eine Alarmzeit ist ebenfalls einstellbar durch ein USR-Kommando, in dem vor der Zeiteingabe noch ein Buchstabe steht. Beispielsweise stellt A = USR(»A1200000«) einen Wecker auf 12 Uhr. Der Alarm im Programm läßt den Bildschirmrahmen blinken. Abstellen kann man das durch Auslösen eines RESTORE-NMI (also RUN/STOP und RESTORE). Sollten Sie vor dem eingestellten Alarm mal solch einen NMI auslösen, dann muß die Alarmzeit neu gestellt werden. Als Basis für dieses Programm diente ein Listing aus dem schon oft erwähnten Buch von Babel/Krause/Dripke »Das Interface Age Systemhandbuch zum Commodore 64«.

Die Unterbrechungs-Programmierung ist damit abgeschlossen — ebenso diese Serie, die als Einführung in die Assembler-Alchimie nun alle Geheimnisse der Kunst aufgedeckt hat. In den letzten Folgen sind wir schon in die Meistergrade der Zunft aufgestiegen. Vielleicht ging es manchem etwas zu schnell? Dann wird Ihnen der anschließende Kurs »Von Basic zu Assembler« eine Hilfe sein, der behutsam und mit vielen an Basic angelehnten Beispielen die nötige Programmierpraxis vermitteln wird. So wie die Segler sich oft »Mast- und Schotbruch« wünschen, verabschiede ich mich, indem ich Ihnen viele grandiose Abstürze wünsche.

(Heimo Ponnath/gk)

```
PASS 2
7000           0823 ;
7000           084C ;***************************************
7000           0875 ;*                                     *
7000           089E ;*  ECHTZEITUHR MIT ALARMFUNKTION      *
7000           08C7 ;*                                     *
7000           08F0 ;*  LAEUFT MIT DEM NMI-CIA             *
7000           0919 ;*  IN VERBINDUNG MIT DEM IRQ FUER     *
7000           0942 ;*  DEN ALARM                          *
7000           096B ;*                                     *
7000           0994 ;*    HEIMO PONNATH  HAMBURG  1985     *
7000           09BD ;*                                     *
7000           09E6 ;*  (TEILWEISE WURDE EIN PROGRAMM AUS  *
7000           0A0F ;*  DEM INTERFACE AGE SYSTEMHANDBUCH   *
7000           0A38 ;*  ZUM COMMODORE 64 , SEITE 114       *
7000           0A61 ;*  ALS BASIS VERWENDET )              *
7000           0A8A ;*                                     *
7000           0AB3 ;***************************************
7000           0AB6 ;
C000           0AC2                .BA $C000
C000           0AC8                .OS
C000           0ACB ;
C000           0AF4 ;********** ZEROPAGE-LABELS ***********
C000           0AF7 ;
C000           0B18 VERZ           .DE $02              ;AKTUELLE VERZOEG.
C000           0B39 FARB           .DE $03              ;WERT FUER RAHMEN
C000           0B57 ;                EOR-OPERATION
C000           0B78 VORW           .DE $04              ;VERZOEGERUNGSWERT
C000           0B9B VALTYP         .DE $0D              ;INHALT:FF=STR 0=N
C000           0B9E ;
C000           0BAD INDEX          .DE $22
C000           0BC6 INDEX3         .DE $24              ;POINTER
C000           0BD6 INDEX4         .DE $25
C000           0BF5 FAC1           .DE $62              ;1.MANTISSENBYTE
C000           0BF8 ;
C000           0C21 ;********** LABELS PAGES 3 ************
C000           0C24 ;
C000           0C44 USRADDL        .DE $0311            ;USR-POINTER
C000           0C57 USRADDH        .DE $0312
C000           0C5A ;
C000           0C6A FREI           .DE $0313
C000           0C6D ;
C000           0C8A IRQVL          .DE $0314            ;IRQ-VEKTOR
C000           0C9B IRQVH          .DE $0315
C000           0C9E ;
C000           0CBC NMINVL         .DE $0318            ;NMI-VEKTOR
C000           0CCE NMINVH         .DE $0319
C000           0CD1 ;
C000           0CFA ;********** LABELS INTERPRETER ********
C000           0CFD ;
C000           0D23 ILLQUERR       .DE $B248            ;ILLEGAL QUANTITY
C000           0D4B ;            ERROR=NORMALWERT USR-VEKTOR
C000           0D4E ;
C000           0D70 STRINI0        .DE $B47D            ;SPEICHERPLATZ
C000           0D99 ;            PRUEFEN,STRINGPOINTER SETZEN
C000           0DBF STRLIT67       .DE $B4CA            ;REST DER STRING-
C000           0DD8 ;            LESE-ROUTINE
C000           0DF7 LEN1           .DE $B782            ;STRINGLAENGE
C000           0E11 ;            IN Y-REGISTER
C000           0E32 ACTOFC         .DE $BC3C            ;AKKU NACH FAC
C000           0E35 ;
C000           0E5E ;********** LABELS VIC-II-CHIP ********
C000           0E61 ;
C000           0E7E RAND           .DE $D020            ;RAHMENFARBE
C000           0E81 ;
C000           0EAA ;********** LABELS CIA-BAUSTEINE ******
C000           0EAD ;
C000           0ECA CIA1           .DE $DC00            ;START CIA-1
C000           0EEC ICR1           .DE $DC0D            ;IRQ-KONTROLLREG.
C000           0F0E CRA1           .DE $DC0E            ;TIMER-A KONTRREG
C000           0F11 ;
C000           0F34 TOD10TH2       .DE $DD08            ;1/10 SEKUNDEN
C000           0F51 TODSEC2        .DE $DD09            ;SEKUNDEN
C000           0F6D TODMIN2        .DE $DD0A            ;MINUTEN
C000           0F8F TODHR          .DE $DD0B            ;STUNDEN + AM/PM
C000           0FB1 ICR2           .DE $DD0D            ;NMI-KONTROLLREG.
C000           0FD3 CRA2           .DE $DD0E            ;TIMER-A KONTRREG
C000           0FF5 CRB2           .DE $DD0F            ;TIMER-B KONTRREG
C000           0FF8 ;
C000           0FFB ;
C000           1024 ;********** LABELS OBERES ROM **********
C000           1027 ;
C000           1045 NORM           .DE $EA31            ;NORMALER IRQ
C000           1048 ;
C000           106B TASTFLAG       .DE $F6BC            ;TEIL DER NMI-
C000           108E ;               ROUTINE (KEIN MODUL)
C000           10AD VECTAB         .DE $FD2F            ;TABELLE DER
C000           10C8 ;               ROM-VEKTOREN
C000           10E9 VECTAB7        .DE $FD35            ;MSB DES NMI-
C000           110E ;               VEKTORS IN DER TABELLE
C000           1132 IORESET19      .DE $FDB6            ;I/O-RESET:BEI
C000           1157 ;               SETZEN DES CRA IRQ-CIA
C000           117B NMIXCT16       .DE $FE6C            ;NMI-ROUTINE AB
C000           119D ;               SCREEN-EDITOR-RESET
C000           11C1 NMIRS232       .DE $FE72            ;NMI-ROUTINE AB
C000           11DE ;               RS232-HANDLING
C000           11FF NMIEND         .DE $FEBC            ;ENDE DER NMI-
C000           1215 ;               ROUTINE
C000           1237 STOP           .DE $FFE1            ;KERNAL STOP SPRG
C000           1254 ;               NACH JMP($328)
C000           1257 ;
C000           125A ;
C000           1283 ;********** AKTIVIEREN ****************
C000           1286 ;
C000 A9 8E     12A2 INIT           LDA #L,USR           ;USR-VEKTOR
C002 8D 11 03  12B7                STA USRADDL          ;LADEN
C005 A9 C0     12C4                LDA #H,USR
C007 8D 12 03  12D2                STA USRADDH
C00A           12D5 ;
C00A A9 1D     12F2                LDA #L,NMI           ;NMI-VEKTOR MIT
C00C 8D 18 03  130D                STA NMINVL           ;STARTADRESSE
C00F A9 C0     1327                LDA #H,NMI           ;DER EIGENEN
C011 8D 19 03  1345                STA NMINVH           ;NMI-ROUTINE LAD
C014           1348 ;
C014 AD 0E DD  1365                LDA CRA2             ;BIT7 CRA SETZEN:
C017 09 80     137C                ORA #$80             ;%1000 0000
C019 8D 0E DD  1397                STA CRA2             ;NETZFREQ.=50HZ
C01C           139A ;
C01C 60        13A0                RTS
C01D           13A3 ;
C01D           13CC ;********** EIGENE NMI-ROUTINE ********
C01D           13CF ;
C01D 48        13EF NMI            PHA                  ;ANFANG NORMALE NMI-R.
C01E 8A        1406                TXA                  ;REGISTER RETTEN
C01F 48        140C                PHA
C020 98        1412                TYA
C021 48        1418                PHA
C022           141B ;
C022 A9 7F     1439                LDA #$7F             ;SPERREN ALLER NMI
C024 8D 0D DD  1444                STA ICR2
C027           1447 ;
C027 AC 0D DD  1462                LDY ICR2             ;PRUEFEN OB NMI
C02A 10 06     1481                BPL RESTNMI          ;VOM CIA2 KOMMT.
C02C           14A4 ;                 WENN NEIN=SPRUNG
C02C 4C 6A C1  14C1                JMP ALARM            ;WENN JA, ALARM
C02F 4C 72 FE  14E1 CIANMI         JMP NMIRS232         ;REST DER
C032           1508 ;                 NORMALEN NMI-ROUTINE
C032           150B ;
C032           1534 ;***** EIGENE RESTORE-NMI-ROUTINE *****
C032           1537 ;
C032           155F ;   DIE MODULPRUEFUNG WIRD AUSGELASSEN
C032           1562 ;
C032 20 BC F6  1587 RESTNMI        JSR TASTFLAG         ;TEIL DER NMI-
C035 20 E1 FF  15A5                JSR STOP             ;ROUTINE ZUR STOP-
C038 D0 F5     15C2                BNE CIANMI           ;TASTEN-ABFRAGE
C03A           15C5 ;
C03A A2 04     15E3                LDX #$04             ;IRQ UND BRK VEKT.
C03C BD 2F FD  1606 UMLAD1         LDA VECTAB,X         ;RESTAURIEREN
C03F 9D 13 03  1613                STA FREI,X
C042 CA        1619                DEX
C043 D0 F7     1626                BNE UMLAD1
C045           1629 ;
C045           1651 ;   DER NMI-VEKTOR WIRD UEBERSPRUNGEN
C045           1654 ;
C045 A2 1A     1671                LDX #$1A             ;RESTAURIEREN DER
C047 BD 35 FD  1693 UMLAD2         LDA VECTAB7,X        ;RESTLICHEN
C04A 9D 19 03  16AC                STA NMINVH,X         ;VEKTOREN
C04D CA        16B2                DEX
C04E D0 F7     16BF                BNE UMLAD2
C050           16C2 ;
C050           16E3 ;  ZUNAECHST NORMALER I/O-RESET
C050           16E6 ;
C050 A9 7F     16FD                LDA #$7F             ;= 0111 111
C052 8D 0D DC  171B                STA ICR1             ;SPERREN ALLER IRQ
C055 8D 0D DD  1739                STA ICR2             ;SPERREN ALLER NMI
C058 8D 00 DC  1753                STA CIA1             ;DATENREGISTER
C05B           177A ;                 PORT A AUF NORMALWERT
C05B A9 08     1791                LDA #$08             ;=0000 1000
C05D 8D 0E DC  17AD                STA CRA1             ;TIMER A IM CIA1
C060           17B0 ;
C060           17D2 ;  ERSATZ FUER BELEGUNG DES CRA2
C060           17D5 ;
C060 A9 88     17EC                LDA #$88             ;=1000 1000
C062 8D 0E DD  1809                STA CRA2             ;TIMER A IM CIA2:
C065           1829 ;                 BIT 0 AUF STOP
C065           184F ;                 BIT 3 AUF EINZELLAUF
C065           1875 ;                 BIT 5 SYSTEMTAKT EIN
C065           189D ;                 BIT 7 ECHTZEITUHR=50HZ
C065           18A0 ;
C065           18C0 ;  REST DES NORMALEN I/O-RESET
C065           18C3 ;
C065           18C6 ;
C065 A9 08     18DE                LDA #$08             ;= 0000 1000
C067 20 B6 FD  18EE                JSR IORESET19
C06A           18F1 ;
C06A           1919 ;  REST DER NORMALEN RESTORE-NMI-ROUT.
C06A           191C ;
C06A 4C 7C     193A                JMP NMIXCT16         ;EINSPRUNG BEI
C06D           195F ;                 SCREEN EDITOR RESET
C06D           1962 ;
C06D           198B ;**** ABSCHALTEN DER TIME OF DAY UHR **
C06D           198E ;
C06D           19AD ;           DURCH SYS-KOMMANDO
C06D           19B0 ;
C06D A9 48     19D1 AUS            LDA #L,ILLQUERR      ;USR-VEKTOR
C06F 8D 11 03  19EF                STA USRADDL          ;AUF NORMALWERT
C072 A9 B2     1A01                LDA #H,ILLQUERR
C074 8D 12 03  1A0F                STA USRADDH
C077           1A12 ;
C077 78        1A18                SEI
C078 A9 47     1A35                LDA #$47             ;RESTAURIEREN DES
C07A 8D 18 03  1A4F                STA NMINVL           ;NMI-VEKTORS
C07D A9 FE     1A5A                LDA #$FE
C07F 8D 19 03  1A67                STA NMINVH
C082           1A6A ;
C082 A9 31     1A87                LDA #L,NORM          ;RESTAURIEREN
C084 8D 14 03  1AA4                STA IRQVL            ;DES IRQ-VEKTORS
C087 A9 EA     1AB2                LDA #H,NORM
C089 8D 15 03  1ABE                STA IRQVH
C08C           1AC1 ;
C08C 58        1AC7                CLI
C08D 60        1ACD                RTS
C08E           1AD0 ;
C08E           1AF9 ;**** DURCH USR AUFRUFBARE ROUTINE ****
C08E           1AFC ;
C08E 24 0D     1B1D USR            BIT VALTYP           ;WELCHER TYP VON
C090           1B44 ;                 VARIABLEN LIEGT VOR ?
C090 30 03     1B5F                BMI STRING           ;WENN STRING,
C092           1B93 ;                 DANN UEBERSPRINGEN
C092 4C 20 C1  1B9F                JMP ZAHLVAR          ;SONST SPRUNG
C095           1BA2 ;
C095           1BCB ;****** STELLEN DER ECHTZEITUHR *******
C095           1BEB ;           DURCH USR("HHMMSST")
C095           1BEE ;
C095 20 92 B7  1C0F STRING         JSR LEN1             ;Y=STRINGLAENGE
C098 C0 07     1C2C                CPY #$07             ;STRING=7ZEICHEN?
C09A D0 40     1C4A                BNE ALSET            ;NEIN DANN ALARM
C09C           1C69 ;                 ZEIT STELLEN?
C09C           1C6C ;
C09C AD 0F DD  1C88                LDA CRB2             ;TIMER B IN CIA2
C09F 29 7F     1CA3                AND #$7F             ;BIT7 LOESCHEN:
C0A1 8D 0F DD  1CBF                STA CRB2             ;NORMALE UHRZEIT
C0A4           1CE4 ;                 IN ECHTZEITUHR CIA2
C0A4           1CE7 ;
C0A4           1D05 ;  AUSLESEN DES ZEIT-STRINGS
C0A4           1D08 ;
C0A4 A0 00     1D22                LDY #$00             ;ZAEHLER AUF 0
C0A6 A9 24     1D47 STELLEN        LDA #$24             ;BCD 24 STD-VERGL.
C0A8 20 FE C0  1D65                JSR ASCBCD           ;ZEICHENTEST UND
C0AB           1D8D ;                 UMWANDLUNG IN BCD-ZAHL
C0AB D0 02     1DAB                BNE STD12            ;STUNDEN UNGLEICH
C0AD           1DCF ;                 NULL ? DANN SPRUNG
C0AD A9 24     1DE6                LDA #$24             ;SONST = 24
C0AF           1DE9 ;
C0AF C9 13     1E08 STD12          CMP #$13             ;STUNDEN GROESSER
C0B1 90 07     1E2A                BCC STDSET           ;ODER GLEICH 12 ?
```

Listing 4. Der Quelltext zur Echtzeituhr

```
C0B3           1E4D ;                NEIN, DANN SPRUNG
C0B3 F8        1E67                  SED                 ;SONST DAVON BCD 12
C0B4 38        1E7B                  SEC                 ;SUBTRAHIEREN
C0B5 E9 12     1E8B                  SBC #$12            ;UND
C0B7 D8        1E91                  CLD
C0B8 09 80     1EA9                  ORA #$80            ;BIT7 SETZEN
C0BA           1EAC ;
C0BA 8D 0B DD  1ECF STDSET           STA TODHR           ;BCD-STUNDEN UND
C0BD           1EF7 ;                AM/PM-FLAG IN TOD-CIA2
C0BD           1EFA ;
C0BD 20 FC C0  1F18                  JSR ASCBCD1         ;ZEICHENTEST U.
C0C0           1F3F ;                UMWANDELN IN BCD-ZAHL
C0C0 8D 0A DD  1F5A                  STA TODMIN2         ;ERGEBNIS IN
C0C3           1F7F ;                TOD-MINUTENREGISTER
C0C3           1F82 ;
C0C3 20 FC C0  1F9F                  JSR ASCBCD1         ;DASSELBE FUER
C0C6 8D 09 DD  1FBB                  STA TODSEC2         ;DIE SEKUNDEN
C0C9           1FBE ;
C0C9 20 66 C1  1FDA                  JSR TEST            ;PRUEFEN,OB 1/10
C0CC           1FF9 ;                SEKUNDEN=ZAHL
C0CC 8D 08 DD  2017                  STA TOD10TH2        ;UND EINTRAGEN
C0CF           2039 ;                INS TOD-REGISTER
C0CF           205D ;  DIE UHR BEGINNT JETZT ZU LAUFEN
C0CF           2060 ;
C0CF A9 00     207D                  LDA #$00            ;KENNUNG FUER OK.
C0D1 4C 3C BC  20A2 AKKUFAC          JMP ACTOFC          ;AKKU ZUR UEBER-
C0D4           20C9 ;                GABE INS BASIC IN FAC
C0D4           20CC ;
C0D4           20F5 ;********** FEHLER AUFGETRETEN ********
C0D4           20F8 ;
C0D4 68        2116 FEHLER           PLA                 ;JSR-ADRESSEN VOM
C0D5 68        212A                  PLA                 ;STAPEL HOLEN
C0D6           212D ;
C0D6 68        2138 ERROR            PLA
C0D7 68        213E                  PLA
C0D8           2141 ;
C0D8 A9 FF     2164 ERROR1           LDA #$FF            ;FEHLERKENNUNG IN
C0DA D0 F5     2180                  BNE AKKUFAC         ;AKKU UND FAC
C0DC           2183 ;
C0DC           21AC ;ENDE DIESES TEILS D. UNBEDINGTEN SPRG.
C0DC           21D5 ;---------------------------------------
C0DC           21D8 ;
C0DC           2201 ;**** ALARMZEIT EINLESEN **************
C0DC           2204 ;
C0DC           222A ;  AUFRUF DURCH Z.B. USR("AHHMMSST")
C0DC           222D ;
C0DC C0 08     224B ALSET            CPY #$08            ; 8 ZEICHEN ?
C0DE D0 F8     2265                  BNE ERROR1          ;NEIN=FEHLER
C0E0           2268 ;
C0E0 AD 0F DD  2273                  LDA CRB2
C0E3 09 80     228E                  ORA #%10000000      ;ALARMBIT
C0E5 8D 0F DD  22A1                  STA CRB2            ;SETZEN
C0E8           22A4 ;
C0E8 A9 84     22C0                  LDA #%10000100      ;ALARM-NMI
C0EA 8D 0D DD  22D5                  STA ICR2            ;ZULASSEN
C0ED           22D8 ;
C0ED A9 3C     22F3                  LDA #$3C            ;VERZOEGERUNGS-
C0EF 85 04     230D                  STA VORW            ;WERT VORGEBEN
C0F1 85 02     2318                  STA VERZ
C0F3 A9 FF     2336                  LDA #$FF            ;EOR-WERT VORGEBEN
C0F5 85 03     2341                  STA FARB
C0F7 A0 01     235F                  LDY #$01            ;BUCHSTABE UEBERL.
C0F9 4C A6 C0  236D                  JMP STELLEN
C0FC           2370 ;
C0FC           2373 ;
C0FC           239C ;*************************************
C0FC           23C5 ;UNTERPROGRAMM ZUR UMWANDLUNG DER ASCII
C0FC           23EC ;CODES IN BCD-ZAHLEN UND PRUEFUNG DER
C0FC           23FF ;EINGABE-ZEICHEN.
C0FC           2402 ;
C0FC A9 60     2427 ASCBCD1          LDA #$60            ;BCD 60 ALS GRENZE
C0FE           244F ;                FUER MIN UND SEK WERTE
C0FE           2452 ;
C0FE 85 24     2465 ASCBCD           STA INDEX3
C100 20 13 C1  2482                  JSR TEST1           ;PRUEFEN OB ZAHL
C103 0A        249A                  ASL                 ;AUS LSB INS MSB
C104 0A        24AB                  ASL                 ;SCHIEBEN
C105 0A        24B2                  ASL
C106 0A        24B9                  ASL
C107 85 25     24D7                  STA INDEX4          ;UND ZW.SPEICHER
C109           24DA ;
C109 20 13 C1  24F7                  JSR TEST1           ;NAECHSTE ZIFFER
C10C           2510 ;                PRUEFEN
C10C 05 25     252C                  ORA INDEX4          ;MSB AUS ZWSP.
C10E           2552 ;                UND LSB ZUSAMMENOREN
C10E C5 24     256F                  CMP INDEX3          ;UNTER GRENZW.?
C110 B0 C4     258D                  BCS ERROR           ;NEIN=FEHLERAUSG.
C112           2590 ;
C112 60        2596                  RTS
C113           2599 ;
C113           25C2 ;** PRUEFUNG OB ASCII-ZAHL VORLIEGT ***
C113           25C5 ;
C113 B1 22     25E8 TEST1            LDA (INDEX),Y       ;ZEICHEN EIN-
C115           2607 ;                LESEN IN AKKU
C115 38        260D                  SEC
C116 E9 30     2625                  SBC #$30            ;< ASCII 0 ?
C118 90 BA     263D                  BCC FEHLER          ;JA=FEHLER
C11A           2640 ;
C11A C9 0A     2659                  CMP #$0A            ;>= ASCII : ?
C11C B0 B6     2671                  BCS FEHLER          ;JA=FEHLER
C11E           2674 ;
C11E C8        2690                  INY                 ;SCHLEIFENZAEHLER + 1
C11F 60        2696                  RTS
C120           2699 ;
C120           26C2 ;**** ENDE PROGRAMMTEIL UHR STELLEN ***
C120           26C5 ;
C120           26EE ;---------------------------------------
C120           26F1 ;
C120           271A ;********** UHR LESEN *****************
C120           271D ;
C120           273E ;     GESCHIEHT DURCH USR(ZAHL)
C120           2741 ;
C120 A9 07     2761 ZAHLVAR          LDA #$07            ;STRINGLAENGE
C122 20 7D B4  277F                  JSR STRINI8         ;SCHAFFT 7 BYTE
C125           27A7 ;                PLATZ FUER STRING UND
C125           27CF ;                LEGT START NACH $62/63
C125           27F6 ;                SOWIE LAENGE NACH $61
C125           2814 ;                (FAC $61-66)
C125           2817 ;
C125 A0 00     2831                  LDY #$00            ;ZAEHLER AUF 0
C127 AD 0B DD  284F                  LDA TODHR           ;STUNDE AUSLESEN,
C12A           2878 ;                DABEI WIRD GESAMTE ZEIT
C12A           28A1 ;                ZWISCHENGESPEICHERT UND
C12A           28C7 ;                ERST NACH LESEN DER
C12A           28EF ;                1/10-SEK ZURUECKGEHOLT
C12A           2914 ;                MIT AKTUELLEN WERT.
C12A           2917 ;
C12A 08        2935                  PHP                 ;STATUS ZWISCHENSPEICH.
C12B           2938 ;
C12B 29 1F     294F                  AND #$1F            ;=0001 1111
C12D           2978 ;                LOESCHEN DER AM/PM-FLAG
C12D C9 12     2996                  CMP #$12            ;=0001 0010 =BCD12
C12F D0 02     29B0                  BNE NO12            ;<>DANN SPRUNG
C131 A9 00     29CE                  LDA #$00            ;SONST STATTDESSEN
C133           29D1 ;
C133 28        29F0 NO12             PLP                 ;STATUS ZURUECKHOLEN
C134 10 05     2A0A                  BPL AM              ;FALLS KEINE AM/
C136           2A2F ;                PM-FLAG GESETZT WAR
C136           2A32 ;
C136 F8        2A4C                  SED                 ;SONST ADDIEREN VON
C137 18        2A62                  CLC                 ;BCD 12 WEIL PM
C138 69 12     2A6D                  ADC #$12
C13A D8        2A73                  CLD
C13B           2A76 ;
C13B 20 55 C1  2A96 AM               JSR BCDASC          ;UP ZUR UMRECHNG
C13E           2ABC ;                VOM BCD IN ASCII UND
C13E           2AE5 ;                ABLEGEN IM STRING. HIER
C13E           2B02 ;                STUNDENWERT
C13E           2B05 ;
C13E AD 0A DD  2B22                  LDA TODMIN2         ;DASSELBE FUER
C141 20 55 C1  2B3C                  JSR BCDASC          ;MINUTENWERT
C144           2B3F ;
C144 AD 09 DD  2B5C                  LDA TODSEC2         ;UND SEKUNDEN-
C147 20 55 C1  2B6F                  JSR BCDASC          ;WERT
C14A           2B72 ;
C14A AD 08 DD  2B9C                  LDA TOD10TH2        ;UND 1/10-
C14D 20 60 C1  2BA2                  JSR BCDASC1         ;SEKUNDENWERT
C150           2BAB ;
C150 68        2BC7                  PLA                 ;USR-STRING-ARGUMENT-
C151 68        2BE5                  PLA                 ;RUECKSPRUNG VORBEREIT.
C152           2BE8 ;
C152 4C CA B4  2C06                  JMP STRLIT67        ;BRINGT STRING
C155           2C29 ;                DESCRIPTOR IN DIE
C155           2C52 ;                DESCRIPTORTABELLE ($19-
C155           2C79 ;                $21),SETZT POINTER IN
C155           2CA2 ;                FAC(HIER $64/65) DARAUF
C155           2CC9 ;                SETZT STRING-FLAGGE,
C155           2CE9 ;                ERHOEHT LETZTEN
C155           2D10 ;                DESCRIPTOR-INDEX UM 3
C155           2D38 ;                ROUTINE ENDET MIT RTS.
C155           2D3B ;
C155           2D64 ;**** ENDE DES LESENS DER UHR *********
C155           2D67 ;
C155           2D90 ;---------------------------------------
C155           2D93 ;
C155           2DBC ;* UNTERPROGRAMM Z. UMRECHNUNG BCD IN *
C155           2DE5 ;* ASCII U.EINTRAGEN IN STRINGSPEICHER*
C155           2DE8 ;
C155 48        2E0B BCDASC           PHA                 ;AUF STAPEL ZW.SPEICH.
C156           2E0E ;
C156 4A        2E2A                  LSR                 ;MSB INS LSB SCHIEBEN
C157 4A        2E31                  LSR
C158 4A        2E38                  LSR
C159 4A        2E3F                  LSR
C15A           2E42 ;
C15A 20 60 C1  2E5E                  JSR BCDASC1         ;IN ASCII UM-
C15D           2E85 ;                RECHNEN UND SPEICHERN
C15D           2E88 ;
C15D 68        2EA5                  PLA                 ;ZURUECKHOLEN DER BCD-
C15E 29 0F     2EC3                  AND #$0F            ;ZAHL,LOESCHEN DES
C160           2ED8 ;                MSB
C160           2EDB ;
C160 09 30     2F00 BCDASC1          ORA #$30            ;DAZUODERN VON $30
C162           2F28 ;                ERZEUGT (WEIL NUR ZAHL
C162           2F4F ;                ZWISCHEN 0 UND 9) DEN
C162           2F78 ;                ASCII-WERT ($30 BIS 39)
C162           2F7B ;
C162 91 62     2F98                  STA (FAC1),Y        ;EINTRAGEN IN
C164           2FB7 ;                STRINGTABELLE
C164           2FBA ;
C164 C8        2FCC                  INY                 ;ZAEHLER +1
C165           2FCF ;
C165 60        2FD5                  RTS
C166           2FD8 ;
C166           3001 ;****** REST DES UP ASCII-BCD *********
C166           3004 ;
C166 20 13 C1  3020 TEST             JSR TEST1           ;PRUEFT AUF
C169           3042 ;                ASCII-ZAHL (0-9)
C169 60        3048                  RTS
C16A           3049 ;
C16A           3074 ;****** NMI-REAKTION AUF ALARM ********
C16A           3077 ;
C16A A9 77     3097 ALARM            LDA #<ALIRQ         ;NEUER IRQ-
C16C 8D 14 03  30AB                  STA IRQVL           ;VEKTOR
C16F A9 C1     30BA                  LDA #>ALIRQ
C171 8D 15 03  30C6                  STA IRQVH
C174           30C9 ;
C174 4C BC FE  30E5                  JMP NMIEND          ;REST DER NOR-
C177           3108 ;                MALEN NMI-ROUTINE
C177           310B ;
C177           3134 ;****** DIE  NEUE IRQ-ROUTINE *********
C177           3151 ;                RAHMENBLINKEN
C177           3154 ;
C177           3157 ;
C177 C6 02     3175 ALIRQ            DEC VERZ            ;ZEITSCHLEIFE
C179 F0 03     3191                  BEQ BLINK           ;BLINKEN WENN 0
C17B           3194 ;
C17B 4C 31 EA  31B2                  JMP NORM            ;SONST NORMALE IRQ
C17E           31B5 ;
C17E A5 04     31D5 BLINK            LDA VORW            ;ZAEHLER RUECK-
C180 85 02     31E8                  STA VERZ            ;SETZEN
C182           31EB ;
C182 AD 20 D0  3203                  LDA RAND            ;RAHMENFARBE
C185 45 03     321B                  EOR FARB            ;INVERTIEREN
C187 8D 20 D0  3226                  STA RAND
C18A           3229 ;
C18A 4C 31 EA  3244                  JMP NORM            ;ZUM NORMAL-IRQ
C18D           3247 ;
C18D           324D                  .EN
```

Listing 4. Der Quelltext zur Echtzeituhr (Schluß)