

```
programm : save-obj          c000 c0d3
```

```
c000 : a2 15 a0 c0 8e 08 03 8c bb
c008 : 09 03 a2 2b a0 c0 8e 8f 0a
c010 : 02 8c 90 02 60 20 73 00 92
c018 : c9 40 f0 06 20 79 00 4c 65
c020 : e7 a7 20 73 00 20 38 c0 b5
c028 : 4c 7b a4 a5 2e cd d3 c0 32
c030 : d0 03 20 38 c0 4c 48 eb f8
c038 : a9 01 a2 08 a0 6f 20 ba 87
c040 : ff a9 00 20 bd ff 20 c0 f6
c048 : ff a9 08 20 b1 ff a9 6f c3
c050 : 20 93 ff a2 00 bd d7 c0 5d
c058 : 20 a8 ff e8 8a cd d6 c0 dd
c060 : d0 f3 a9 08 20 ae ff a9 60
c068 : 01 20 c3 ff ad d3 c0 18 17
c070 : 6d d4 c0 8d d3 c0 a2 00 f7
c078 : bd a7 c0 20 d2 ff e8 8a 23
c080 : c9 2c d0 f4 a9 00 a2 08 67
c088 : 20 ba ff a2 d9 a0 c0 ad 5a
c090 : d5 c0 20 bd ff a9 00 85 dd
c098 : 9d a9 2b a6 2d a4 2e 20 9a
c0a0 : d8 ff a9 80 85 9d 60 11 db
c0a8 : 20 20 20 20 20 12 20 50 98
c0b0 : 52 4f 47 52 41 4d 4d 20 ba
c0b8 : 57 49 52 44 20 41 42 47 75
c0c0 : 45 53 50 45 49 43 48 45 c6
c0c8 : 52 54 20 92 20 20 20 20 63
c0d0 : 20 20 11 85 3b 85 2d 4c 23
```

Listing 1.
»SAVE.OBJ«
wird von
Listing 2
»AUTO-SAVE«
nachgeladen.
Bitte
beachten
Sie die
Eingabe-
hinweise
auf Seite 7.

Ziffern und Zeiger auf dem C 64

Digitaluhren sind out. Um dem Trend der Zeit zu folgen, haben wir für Sie eine »Analog-Uhr« auf dem C64 installiert.

Es sind schon etliche Versionen von Digitaluhren auf dem C64 veröffentlicht worden. Zur Abwechslung haben wir diesmal eine Interrupt-gesteuerte Analoguhr für Sie. Das Programm blendet ein klassisches Ziffernblatt mit Zeigern in die rechte obere Ecke des Bildschirms ein.

Hinweise zum Eintippen. Geben Sie zuerst das Maschinenprogramm »UHR PRG.« (Listing 1) mit dem MSE ein, und speichern Sie es. Dann tippen Sie »UHR SPRITEGEN« (Listing 2) mit dem Checksummer ab und speichern es eben-

falls. Starten Sie »UHR SPRITEGEN«. Dieses lädt zuerst das Maschinenprogramm und berechnet danach die Sprite-Daten für die Uhr und die Zeiger. Nach etwa vier Minuten werden die Spritedaten inklusive der Interruptroutine auf Diskette gespeichert. Das Programm wird dann mit LOAD »ANALOGUHR«, 8,1 geladen.

Zum Programm. Die Uhr wird durch »SYS 40000,hh,mm,ss« eingeschaltet. Dabei bedeuten hh, mm, ss jeweils Stunden, Minuten und Sekunden mit je zwei Ziffern (Zum Beispiel »SYS 40000,15,05,00« stellt die Uhr auf 15:05:00). Der Bildschirmspeicher liegt jetzt von 35840 bis 36839 und der Basic-Speicher wird auf etwa 33000 Bytes begrenzt. Durch RUN/STOP-RESTORE wird die Uhr abgeschaltet und der Bildschirm liegt wieder bei 1024. Das Ziffernblatt und die drei Zeiger bestehen aus Sprites, die von der neuen Interrupt-Routine gesteuert werden. Diese Routine liest bei jedem Interrupt die »time-of-day«-Register der CIA 1, berechnet für jeden Zeiger die Bildschirmposition und setzt die Spritpointer. Dann wird zum normalen Interrupt des Betriebssystems gesprungen.

Programmbeschreibung: Das Basic-Programm »UHR SPRITEGEN« besteht hauptsächlich aus drei Schleifen. In den Zeilen 160-260 werden die Zeigersprites erzeugt. Die IF-Zeilen bestimmen die Startkoordinaten der Linien. Durch READ in Zeile 220 werden die Endkoordinaten aus dem ersten Datenblock gelesen. Die aufgerufenen Unterprogramme löschen das zu bearbeitende Sprite (460), ziehen eine Linie im Sprite (370) und übertragen das Sprite in den endgültigen Speicher unter dem Basic-ROM (480). In den Zeilen 270-290 werden die Daten für die vier Sprites, die das Zifferblatt darstellen, gelesen und gePOKEt. Zeile 300 liest die Daten für das Maschinenprogramm, das das Programm »ANALOGUHR« speichert. Dieses wird in Zeile 310 aufgerufen. Das Unterprogramm Zeile 340-360 setzt einen Punkt im bearbeiteten Sprite, in Zeile 380-440 werden die Punkte für eine Linie (X1, Y1, X2, Y2) berechnet und gesetzt. Zeile 460 löscht das Arbeitssprite. In Zeile 480 werden die Spritedaten in den Speicherbereich, entsprechend der Variablen »W«, die in Zeile 240 berechnet wird, übertragen. Die Datenzeilen enthalten die X2,Y2-Koordinaten der Zeiger, die Daten für das Zifferblatt und für die Speicherroutine.

Das Maschinenprogramm »UHR PRG.« arbeitet folgendermaßen: nach SYS 40000 wird zuerst die Uhr in CIA1 gestellt, der Bildschirm verschoben und die Spriteregister des VIC initialisiert. Dann wird der Basic-Speicher begrenzt und der NMI- und IRQ-Vektor verbogen. Anschließend wird zum Basic-Warmstart gesprungen. Die neue NMI-Routine setzt nur den Bildschirm wieder zurück, dann wird zum »Restore« des Betriebssystems gesprungen.

```
programm : uhr prg.          9c40 9e4c
```

```
9c40 : ad 0f dc 29 7f 8d 0f dc 2b
9c48 : ad 0e dc 09 80 8d 0e dc bb
9c50 : a2 03 d0 03 20 73 00 20 e6
9c58 : fd ae 38 e9 30 0a 0a 0a 87
9c60 : 0a 85 fd 20 73 00 38 e9 9c
9c68 : 30 05 fd c9 60 90 03 4c 03
9c70 : 48 b2 e0 03 d0 0d c9 24 8f
9c78 : b0 f5 c9 12 90 05 18 f8 5b
9c80 : 69 68 d8 9d 08 dc ca d0 3b
9c88 : cb 8e 08 dc a9 34 8d 18 db
9c90 : d0 a7 95 8d 00 dd a7 8c fb
9c98 : 8d 88 02 a9 ff dd 15 d0 81
9ca0 : 8d 10 d0 8e 29 d0 a9 01 fd
9ca8 : 8d 28 d0 8d 2a d0 8d 2b e5
9cb0 : d0 8d 2c d0 8d 2d d0 8d 0d
9cb8 : 27 d0 a9 8c 85 34 85 38 c4
9cc0 : 78 a9 21 8d 14 03 a9 9d 42
9cc8 : 8d 15 03 a9 19 8d 18 03 3a
9cd0 : a9 9d 8d 19 03 68 68 58 94
9cd8 : a9 22 8d 06 d0 8d 0a d0 fa
```

```
9ce0 : a9 3a 8d 0c d0 8d 08 d0 c6
9ce8 : a9 38 8d 07 d0 8d 07 d0 31
9cf0 : a9 4d 8d 0b d0 8d 0d d0 54
9cf8 : a9 d4 8d fb 8f a9 d5 8d a7
9d00 : fc 8f a9 d6 8d fd 8f a9 63
9d08 : d7 8d fe 8f a9 93 20 d2 b5
9d10 : ff a9 80 8d 00 a0 4c 74 d5
9d18 : a4 a9 04 8d 88 02 4c 66 da
9d20 : fe 78 ad 0b dc 29 1f 20 fb
9d28 : 30 9e aa 0a 18 69 bc 85 5e
9d30 : fd 8a c9 09 90 0e ad 4a cb
9d38 : 9e 8d 00 d0 ad 4b 9e 8d 82
9d40 : 01 d0 d0 30 c9 06 90 0e 0f
9d48 : ad 48 9e 8d 00 d0 ad 49 42
9d50 : 9e 8d 01 d0 d0 1e c9 03 3a
9d58 : 90 0e ad 46 9e 8d 00 d0 1b
9d60 : ad 47 9e 8d 01 d0 d0 0c fc
9d68 : ad 44 9e 8d 00 d0 ad 45 58
9d70 : 9e 8d 01 d0 ad 0a dc 20 0e
9d78 : 30 9e aa 09 80 8d f9 8f 3f
9d80 : 8a c9 2d 90 15 ad 4a 9e 71
9d88 : 8d 02 d0 ad 4b 9e 8d 03 e6
9d90 : d0 a9 0f 05 fd 8d 8f 8f 65
```

```
9d98 : d0 41 c9 1e 90 15 ad 48 38
9da0 : 9e 8d 02 d0 ad 49 9e 8d 5a
9da8 : 03 d0 a9 01 05 fd 8d f8 06
9db0 : 8f d0 28 c9 0f 90 13 a5 f8
9db8 : fd 8d f8 8f ad 46 9e 8d 4e
9dc0 : 02 d0 ad 47 9e 8d 03 d0 83
9dc8 : d0 11 ad 44 9e 8d 02 d0 15
9dd0 : ad 45 9e 8d 03 d0 a5 fd c2
9dd8 : 8d f8 8f ad 09 dc 20 30 d3
9de0 : 9e aa 09 80 8d fa 8f 8a 2a
9de8 : c9 2d 90 0e ad 4a 9e 8d f0
9df0 : 04 d0 ad 4b 9e 8d 05 d0 3d
9df8 : d0 30 c9 1e 90 0e ad 48 d7
9e00 : 9e 8d 04 d0 ad 49 9e 8d 3b
9e08 : 05 d0 d0 1e c9 0f 90 0e e1
9e10 : ad 46 9e 8d 04 d0 ad 47 46
9e18 : 9e 8d 05 d0 d0 0c ad 44 85
9e20 : 9e 8d 04 d0 ad 45 9e 8d 3b
9e28 : 05 d0 ad 08 dc 4c 31 ea cc
9e30 : aa 4a 4a 4a 4a 4a 8a 29 42
9e38 : 0f c0 00 f0 06 18 69 0a a0
9e40 : 88 d0 fa 60 36 39 36 4d 9c
9e48 : 22 4d 22 39 01 ff 34 01 a3
```

Listing 1. Das Maschinenprogramm »UHR PRG.« Bitte mit dem MSE eingeben.

```

100 REM *** ANALOGUHR *** <038>
110 REM MARC MICHAELIS <188>
120 REM PFARRHOEHE 11 <132>
130 REM 8507 OBERASBACH <035>
140 REM <202>
150 IF A=0 THEN A=1:LOAD"UHR PRG.",8,1 <165>
160 FOR J=0 TO 83 <161>
170 IF J=0 OR J=60 THEN X1=0:Y1=20 <164>
180 IF J=15 OR J=66 THEN X1=0:Y1=0 <079>
190 IF J=30 OR J=72 THEN X1=20:Y1=0 <050>
200 IF J=45 OR J=78 THEN X1=20:Y1=20 <101>
210 GOSUB 460 <018>
220 READ X2,Y2 <211>
230 GOSUB 370 <046>
240 W=40960+64*J <165>
250 GOSUB 480 <090>
260 NEXT <016>
270 FOR J=46336 TO 46591 <000>
280 READ A:POKE J,A <192>
290 NEXT <046>
300 FOR J=832 TO 880:READ A:POKE J,A:NEXT <234>
310 SYS 832 <236>
320 END <068>
330 REM PLOT <206>
340 Q=Y*3+INT(X/8)+832:Q1=2+(7-(X-INT(X/8) <156>
*8)) <108>
350 POKE Q,PEEK(Q)OR Q1 <164>
360 RETURN <066>
370 REM LINIE ZIEHEN (X1,Y1,X2,Y2) <231>
380 IF ABS(X2-X1)<5 THEN 420 <253>
390 FOR I=X1 TO X2 STEP(X2-X1)/20 <251>
400 Y=INT((Y2-Y1)/(X2-X1)*(I-X1)+Y1) <039>
410 X=INT(I):GOSUB 330:NEXT:RETURN <204>
420 FOR I=Y1 TO Y2 STEP(Y2-Y1)/20 <081>
430 X=INT(((X2-X1)/(Y2-Y1)*(I-Y1)+X1)) <073>
440 Y=INT(I):GOSUB 330:NEXT:RETURN <042>
450 REM LOESCHEN SPRITE <066>
460 FOR I=832 TO 896:POKE I,0:NEXT:RETURN <133>
470 REM UEBERTRAGEN SPRITE
480 FOR I=0 TO 63:POKE W+I,PEEK(832+I):NEXT:RETURN <004>
490 REM DATA FUER ZEIGER ( X UND Y ) <074>
500 DATA 0,0,2,0,4,1,6,1,8,2,10,3,12,4,13, <123>
5,15,7,16,8,17,10,18,12,19,14,19,16 <143>
510 DATA 20,18
520 DATA 20,0,20,2,19,4,19,6,18,8,17,10,16 <243>
,12,15,13,13,15,12,16,10,17,8,18,6,19 <228>
530 DATA 4,19,2,20
540 DATA 20,20,18,20,16,19,14,19,12,18,10, <142>
17,8,16,7,15,5,13,4,12,3,10,2,8,1,6 <188>
550 DATA 1,4,0,2
560 DATA 0,20,0,18,1,16,1,14,2,12,3,10,4,8 <141>
,5,7,7,5,8,4,10,3,12,2,14,1,16,1,18,0 <078>
570 DATA 0,10,3,10,5,11,7,13,9,15,10,17 <209>
580 DATA 10,0,10,3,9,5,7,7,5,9,3,10 <214>
590 DATA 20,10,17,10,15,9,13,7,11,5,10,3
600 DATA 10,20,10,17,11,15,13,13,15,11,17, <154>
10 <117>
610 REM ZIFFERBLATT SPRITES
620 DATA 0,0,255,0,7,8,0,24,8,0,32,0,0,224 <075>
,85,1,16,37,2,0,85,4,84,0,4,36,0,8
630 DATA 84,0,8,0,0,24,0,0,37,64,0,32,128, <142>
0,65,64,0,64,0,0,64,0,0,128,0,0,128
640 DATA 0,0,128,0,0,138,128,0,239,128,0,0, <050>
,112,0,0,12,0,0,2,0,0,3,128,0,4,64
650 DATA 0,0,32,0,16,16,0,16,8,0,16,4,0,0, <074>
4,0,0,6,0,0,169,0,0,161,0,0,160,128
660 DATA 0,0,128,0,0,128,0,0,64,0,0,64,0,0, <230>
,64,0,84,64,200,233,0,8,138,128,0
670 DATA 128,0,0,128,0,0,128,0,0,85,80,0,8 <198>
5,80,0,73,80,0,32,0,0,36,0,0,24,0
680 DATA 0,8,170,0,8,170,0,4,74,0,2,0,42,1 <035>
,16,42,0,224,18,0,32,0,0,24,8,0,7
690 DATA 8,0,0,255,141,0,85,192,0,84,64,0, <068>
0,64,0,0,64,0,0,64,2,160,128,2,160
700 DATA 128,2,64,128,0,1,0,0,9,0,0,6,0,40 <104>
,4,0,40,4,0,16,24,0,0,32,0,4,64,0
710 DATA 3,128,0,2,0,0,12,0,0,112,0,0,128, <048>
0,0,0 <032>
720 REM DATEN FUER ABSPEICHERROUTINE
730 DATA 169,54,133,1,162,8,32,186,255,169 <168>
,9,162,104,160,3,32,189,255,169,64
740 DATA 133,251,169,156,133,252,169,251,1 <008>
62,0,160,182,32,216,255,169,55,133 <088>
750 DATA 1,96,65,78,65,76,79,71,85,72,82

```

© 64'er

Listing 2. Das Basic-Programm »UHR SPRITEGEN« erzeugt die nötigen Sprites der Uhr, verbindet diese mit dem Maschinenprogramm und speichert beides als »ANALOGUHR« auf Diskette

Das eigentliche Steuerprogramm ist eine neue Interrupt-routine. Zuerst wird die Stunde gelesen, von BCD in binär umgewandelt, und daraus die vorläufige Spritenummer für den Stundenzeiger berechnet. Außerdem wird die Spriteposition des Stundenzeigers berechnet (Stunde < 3 $\triangleq$ rechtes oberes Viertel, Stunde < 6 $\triangleq$ rechtes unteres Viertel etc.). Dann werden die Minuten gelesen, umgerechnet und die Position des Minutenzeigers berechnet. Wenn die Minuten größer oder gleich 30 sind, wird der Stundenzeiger um eine halbe Stunde vorgerückt. Schließlich wird noch der Sekundenzeiger nach der gleichen Methode gesetzt. Bevor zur IRQ-Routine des Betriebssystems gesprungen wird, muß noch das Zehntelsekunden-Register gelesen werden, um die Uhr wieder freizugeben.

(M. Michaelis/og)

Löschen ohne Verluste

Ein dimensioniertes Feld läßt sich normalerweise nicht ohne den Verlust aller Variablenwerte aus dem Speicher entfernen. Dieses Programm macht's möglich.

Ein einmal angelegtes Variablenfeld läßt sich beim Commodore 64 nicht wieder löschen; es sei denn, man benutzt den Befehl CLR, der dann aber sämtliche Variablen löscht. Mit dem Programm »CLEAR« (Listing 1) können Sie ein einzelnes Variablenfeld löschen. Somit ist es möglich, diese Variable neu zu dimensionieren. Mit SYS 49152 wird der CLR-Befehl erweitert. Folgt nun nach CLR ein Variablenname (zum Beispiel CLR A\$), so wird diese Variable (falls vorhanden) aus dem Array-Bereich entfernt. Die Funktionsweise des Programms kann dem Assemblerlisting entnommen werden (Listing 2).

Wie verwaltet der Computer nun ein Variablenfeld? Im Bild rechts sehen Sie, wie die drei möglichen Variablentypen im Speicher abgelegt werden. In den ersten beiden Bytes steht der Variablenname. Besteht der Name aus nur einem Buchstaben, so wird für den zweiten Namen der Wert 0 eingesetzt. Die drei Variablentypen erkennt der Computer daran, das bei einer Stringvariablen im zweiten Byte das Bit 7 gesetzt ist. Zu dem ASCII-Wert des Variablennamens wird also der Wert 128 dazugezählt. Bei einer Intervariablen ist im ersten und zweiten Byte das Bit 7 gesetzt. In den nächsten beiden Bytes steht die Anzahl der Bytes, die das Feld benötigt. Danach folgt die Anzahl der Dimensionen und die Anzahl der einzelnen Felder. Zum Schluß finden wir noch die Variablen selbst. Wie im Bild zu sehen ist, belegt eine Stringvariable 3 Byte, eine Intervariable 2 Byte und eine Fließkommavariablen 5 Byte. Man kann also Speicherplatz sparen, wenn nach Möglichkeit Intervariablen benutzt werden. Dies ist aber nur bei Variablenfeldern der Fall, weil bei nichtdimensionierten Variablen immer 7 Byte pro Variable belegt werden.

(H. Kunz/og)