

DCF - Computer - Schaltuhr

Digitaluhren sind heutzutage nichts Besonderes mehr. Eine Digitaluhr, die nicht mehr per Hand gesetzt werden muß, ist dagegen durchaus angenehm. Man braucht die Uhr übrigens auch nach einem Netzausfall nicht neu zu stellen, obwohl sie ans Netz angeschlossen ist. Außerdem besitzt diese Uhr vier Schaltausgänge, die ganz nach Belieben über eine Woche programmiert werden können. Das "Gehirn" der Schaltung ist ein Mikrocomputer – fast überflüssig, darauf hinzuweisen. Dieser Computer setzt das kodierte DCF-77-Zeitsignal so um, daß man Zeit, Datum und alle Schaltzeiten auf dem Display im Blick hat.

Zeit – drahtlos

Das Wort "Computer" mag in Zusammenhang mit einer Uhr etwas angeberisch klingen. Was diese Uhr allerdings zu bieten hat, ist fast nicht mehr mit einer "normalen" Digitaluhr zu vergleichen! Das kann man schon an der folgenden Beschreibung erkennen. Von einer Uhr erwartet man selbstverständlich, daß sie die "richtige" Zeit anzeigt. Doch was ist richtig? Wenn ein Quarz oder das Lichtnetz als Referenz verwendet werden, gibt es offensichtlich zu viele Unrichtigkeiten.

Die einmal eingestellte Zeit bleibt zwar über lange Zeit genau. Hin und wieder muß man sie aber nachstellen. Man denke nur an die Umstellung von Sommer- auf Winterzeit und umgekehrt.

Die einzig "richtige" Zeit ist aber (für die Bundesrepublik) die "Gesetzliche Zeit für die Bundesrepublik Deutschland", die die Physikalisch-Technische Bundesanstalt (PTB) über den Langwellensender DCF 77 verbreitet. Im Oktoberheft 1980 steht im Artikel "Normalzeit-Empfänger für DCF 77" alles, was man zu dieser offiziellen Zeit wissen muß. Dort wurde auch gleich der Empfänger für das kodierte Zeitsignal vorgestellt. Fehlt noch die Dekodierung der empfangenen Information und ihre Anzeige auf einem Display. Da das nicht mit einer Handvoll Digital-ICs zu machen ist, haben wir uns hier eines Mikroprozessors bedient. Um aber andererseits mit dem Einsatz eines Mikroprozessors des Guten nicht zuviel zu tun, haben wir seine Fähigkeiten so gut wie möglich genutzt. Das Ergebnis – eine Computer-Schaltuhr. Über vier Ausgänge können elektrische Geräte mittels Relais oder Triac ein- und ausgeschaltet werden.

Was kann die Uhr?

Aufgrund der folgenden Beschreibung kann man sich ein Bild machen, ob die Uhr überhaupt dem entspricht, was von ihr erwartet wird. Erfüllt sie die Erwartungen, kann man den nächsten Schritt wagen – den Nachbau. Der schon erwähnte Normalzeitempfänger wird an den Eingang des Mikroprozessors angeschlossen. Sobald nun die Netzspannung eingeschaltet wird, beginnt die Uhr die Sekunden zu zählen. Gleichzeitig fängt der Prozessor damit an, das Zeit-Signal auszuwerten. Hat der Prozessor das Signal zwei Minuten lang störungsfrei empfangen, erscheint die richtige Zeit auf dem Display. Man kann Stunden, Minuten und Sekunden darauf ablesen. Danach wird die Uhr jede Minute mit der Information aus dem Zeitsignal nachgestellt. Die auf dem Display sichtbaren Sekunden werden von einem internen Quarz-Oszillator abgeleitet. Diese Referenz dient auch zur Anzeige einer Zeit und des Datums, wenn der Empfang des Zeitsignals ausbleibt. Drückt man eine bestimmte Taste, erscheint das aktuelle Datum so lange auf dem Display, bis diese Taste wieder losgelassen wird. "Überflüssige" Nullen werden nicht angezeigt.

Mittels einer Reihe von sieben LEDs wird gleichzeitig neben den Sieben-segmentanzeigen der Wochentag angezeigt. Eine kontinuierlich leuchtende "Wochentag-LED" zeigt darüber hinaus an, daß das Zeitsignal gut empfangen wird. Blinkt die LED, so ist das ein Zeichen für schlechten oder gar keinen Empfang. Ist der Empfang mindestens 2 Minuten lang ausreichend und die Uhr wieder mit dem Sender synchron, leuchtet die LED wieder ununterbrochen.

Außer diesen Eigenschaften besitzt die Uhr alle Merkmale einer Schaltuhr. Vier Schaltausgänge wurden schon erwähnt. Drei dieser Ausgänge kann man so programmieren, daß zwei Ein- und zwei Aus-Schaltzeiten gespeichert sind; man kann auch angeben, an welchem Wochentag die Schaltvorgänge ausgelöst werden sollen. Dabei können die Schaltzeiten auf die Minute genau eingegeben werden. Der vierte Ausgang kann zehn Ein- und zehn Aus-Schaltzeiten verteilt über eine ganze Woche, allerdings nur im Viertelstundenabstand, schalten. Diese ganze Schalterei wird natürlich auch vom Mikroprozessor gesteuert. Er kontrolliert die Programmierung gleichzeitig auf Fehler. Hat man zum Beispiel eine Aus- vor eine Einschaltzeit gelegt, dann gibt die Uhr den Fehler auf dem Display an. Die falsche Programmierung muß erst korrigiert werden, bevor die Zeit wieder auf dem Display erscheint.

Schließlich enthält die Uhr eine Gangreserve mittels NiCd-Akkus. Sie übernehmen bei Netzausfall eine Notversorgung, die programmierten Schaltzeiten bleiben erhalten, und die prozessorinterne Uhr (Zeit und Datum) läuft weiter. Das Display leuchtet in diesem Fall

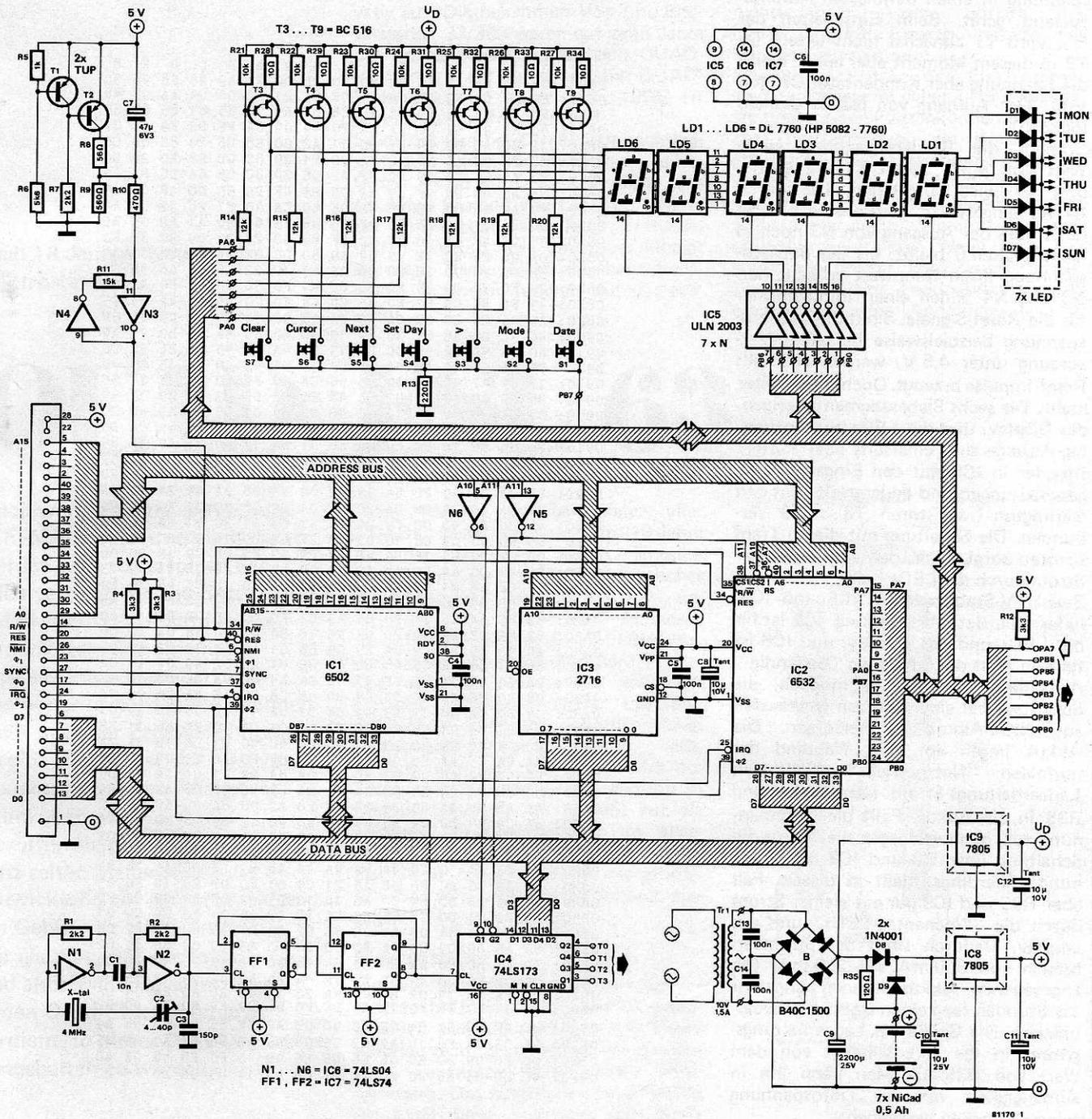


Bild 1. Die Schaltung der Computer-Schaltuhr. Das "Gehirn" ist der 6502-Mikroprozessor. Er wertet die Zeitinformation aus, steuert das Display und die Schaltausgänge.

nur sehr schwach, um Strom zu sparen. An einem Ausgang erscheinen außerdem jede Sekunde Zeit und Datum im ASCII-Code. Die Aufzählung der vielen Möglichkeiten, die die Uhr bietet, zeigt schon, daß der Mikrocomputer über recht effiziente Software verfügt. Der Hardwareaufwand ist im Verhältnis dazu sehr gering.

Hardware

Die Schaltung der Uhr ist in Bild 1 dargestellt. Das "Gehirn" ist der bekannte Mikroprozessor 6502 (IC1). Das Pro-

gramm, die Software, ist in einem 2716-EPROM (IC3) enthalten. IC2 – vom Typ 6532 – enthält 16 Eingänge und Ausgänge, über die das Display angesteuert wird, die Tasten abgefragt werden und die empfangene Zeit-Information aufgenommen wird. Außerdem enthält IC2 einen Timer, der die Sekunden-Impulse erzeugt, wenn die Uhr aufgrund von Sender-Ausfall und/oder Störung weiterlaufen soll. Der zusätzliche 128-Byte-Speicher dient darüber hinaus dazu, Zwischenschritte des Programms und die eingegebenen Schaltzeiten zu speichern.

Außer den 16 Eingangs- und Ausgangsleitungen von IC2 sind weitere 4 Ausgangsleitungen für die verschiedenen Schaltzeiten notwendig. Dazu dient der 4-bit-Zwischenspeicher ("latch") IC4. Links oben im Schaltbild ist der Taktgenerator dargestellt. Der 1-MHz-Takt für den Prozessor wird aus dem Ausgangssignal eines 4-MHz-Quarzoszillators mittels zweier Flipflops (FF1 und FF2) gewonnen. Man kann selbstverständlich auch einen 1-MHz-Quarzoszillator aufbauen. Die hier vorgeschlagene Lösung ist jedoch etwas preiswerter. Mit T1, T2, N3 und N4 wird das RES-

Signal erzeugt. Es sorgt dafür, daß der Prozessor beim Einschalten der Betriebsspannung in einen definierten Anfangszustand gerät. Beim Einschalten der Uhr wird T1 zunächst nicht leiten. Da T2 in diesem Moment aber leitet, bleibt die Spannung über Kondensator C7 Null Volt. Der Ausgang von N3 ist deshalb logisch 0.

Erreicht die Betriebsspannung einen Wert von 4,5 V, leitet T1. T2 sperrt, und C7 wird aufgeladen. Die durch C7/R9 eingestellte Zeitkonstante sorgt dafür, daß der Ausgang von N3 noch so lange logisch 0 bleibt, bis die Betriebsspannung ihren Nennwert erreicht hat. N3 und N4 bilden einen Impulsformer für die Reset-Signale. Sinkt die Betriebsspannung beispielsweise bei Akku-Versorgung unter 4,5 V, werden ebenfalls Reset-Impulse erzeugt. Doch dazu später mehr. Die sechs Siebensegmentanzeigen, das Display, und die LEDs zur Wochentag-Anzeige sind einerseits über Puffer/Inverter in IC5 mit den Eingangs-/Ausgangsleitungen und andererseits mit den Darlington-Transistoren T3...T9 verbunden. Die Schaltung mit diesen Transistoren sorgt dafür, daß ein konstanter Strom durch die LEDs fließt.

Zwei 5-V-Stabilisatoren, IC8 und IC9, liefern die Betriebsspannung. IC9 ist für die LEDs und das Display, und IC8 ist für den Rest der Schaltung "zuständig". Auf diese Weise ist es möglich, die Schaltung bei einem kurzen Netzausfall aus NiCd-Akkus zu versorgen. Die Akkus liegen vor IC8. Während des normalen "Netzbetriebs" fließt ein Ladeerhaltungs-Strom über Widerstand R35 in die Akkus. Fällt die Netzspannung aus, dann versorgen die Akkus die Schaltung über D9 und IC8 mit Spannung. Allerdings fließt in diesem Fall über R35 und IC9 nur ein kleiner Strom durch die "Wochentag-LEDs" und das Display. Dadurch sinkt der Stromverbrauch von 800 mA auf 250 mA. Die angegebenen Akkus können ungefähr 1 1/2 Stunden lang einen Netzausfall überbrücken. Die Größe des Ladeerhaltungsstroms in die Akkus hängt von dem Wert von R35 ab. Man kann ihn in Abhängigkeit von der Trafospaltung folgendermaßen berechnen:

$$R35 = \frac{U_{C9}/V - 10}{20 \cdot 10^{-3}} \Omega.$$

Bleibt die Netzspannung so lange aus, daß die Akkuspannung unter 4,5 V sinkt, dann wird ein Reset-Impuls erzeugt. In diesem Fall blockiert der Reset-Impuls die Schaltung. Dadurch wird verhindert, daß durch Ausfall der Display-Multiplexsteuerung eventuell eine oder mehrere Siebensegmentanzeigen "durchbrennen" können. Selbstverständlich sind dann die programmierten Schaltzeiten verloren gegangen. Glücklicherweise sind solche langen Netzausfälle selten.

Die NiCd-Akkus können auch durch normale Batterien ersetzt werden. Verwendet man beispielsweise zwei in

Tabelle 1. Hex-Dump-Listing des Computer-Schaltuhr-Programms

```

HEXDUMP: 0800,0FFF
0 1 2 3 4 5 6 7 8 9 A B C D E F
0800: D8 A9 FF AA E8 95 00 E0 3D D0 F9 A9 88 E8 95 00
0810: E0 47 D0 F9 A9 00 85 0F 85 21 8D 00 04 AA 95 48
0820: E8 E0 3E D0 F9 E6 4C E6 61 A9 7F 85 81 85 83 85
0830: 60 AA 9A 58 00 EA EA 20 44 09 C9 FF D0 F9 A5 87
0840: A2 00 A0 00 24 87 50 11 A2 00 E8 C8 20 8C 0B C0
0850: 28 F0 5C A5 80 10 F4 30 F1 20 8C 0B E8 E0 28 D0
0860: E3 A5 5F C9 59 D0 3F A2 96 20 8C 0B CA D0 FA A2
0870: 07 A0 06 20 9F 09 A2 05 B5 4F D5 55 D0 1F CA 10
0880: F7 A2 05 B5 55 95 4F 95 49 CA 10 F7 20 32 09 A2
0890: 08 86 61 A2 D8 86 9F 85 48 85 60 30 A1 A2 05 B5
08A0: 55 95 4F CA 10 F9 A2 FF 86 60 20 32 09 30 8F E0
08B0: 0A 90 F3 E0 1E 90 04 A9 80 D0 02 A9 00 85 63 A4
08C0: 5F C0 20 90 2B F0 3C A5 63 10 02 E6 5D A6 5E C0
08D0: 28 F0 38 C0 35 F0 32 C0 41 F0 3C C0 44 F0 42 C0
08E0: 49 F0 45 C0 57 F0 46 C0 58 F0 20 15 55 4A 95 55
08F0: A2 96 20 8C 0B CA D0 FA 18 98 F8 69 01 D8 85 5F
0900: 4C 3E 08 A5 63 30 E9 10 9D 56 55 46 5D B0 97 A9
0910: 00 85 5D E6 5E D0 D9 15 55 4A 4A 95 55 E6 5E D0
0920: CF 15 55 4A 4A 4A 10 F1 15 55 4A 10 EC 15 55 4C
0930: 1B 09 A2 0A A9 00 95 55 CA 10 F9 60 86 5B 84 5C
0940: 46 62 90 12 A2 02 B5 48 E0 02 D0 03 20 89 09 95
0950: 67 CA 10 F2 30 12 A5 4B 20 89 09 85 69 A5 4D 20
0960: 89 09 85 68 A5 4E 85 67 A6 4C A9 7F 38 6A CA D0
0970: FB 85 66 A5 60 F0 0A E6 11 E6 11 10 04 A9 FF 85
0980: 66 20 13 0B A6 5B A4 5C 60 A8 F0 06 C9 0F 80 50
0990: 09 F0 60 A0 00 20 0B 0A 49 60 D0 6E 99 48 00 20
09A0: 0B 0A 49 60 D0 64 99 49 00 20 0B 0A 49 24 D0 5A
09B0: 99 4A 00 20 0B 0A 20 0B 0A C9 08 D0 05 A9 01 99
09C0: 4C 00 B9 4B 00 C9 29 30 41 C9 31 30 11 6A 90 26
09D0: B9 4D 00 C9 08 10 02 49 FF 6A B0 1A 90 2C B9 4D
09E0: 00 C9 02 D0 25 B9 4E 00 29 13 F0 04 C9 12 D0 06
09F0: B9 4B 00 6A B0 14 A9 01 99 4B 00 20 0B 0A C9 13
0A00: D0 08 A9 01 99 4D 00 20 0B 0A 60 18 B5 48 F8 69
0A10: 01 D8 95 48 E8 60 48 98 48 8A 48 A9 00 85 81 A5
0A20: 82 85 64 C6 61 F0 08 A5 01 A9 79 85 9F D0 1B A9
0A30: 82 85 9E A5 1D F0 07 A5 48 D0 03 20 58 0A A2 00
0A40: 20 93 09 20 E9 0A A9 09 85 61 A5 64 85 82 A9 7F
0A50: 85 81 68 AA 68 A8 68 40 85 01 A2 02 A9 00 A8 8A
0A60: 0A 0A 0A 0A 85 0E B1 0E 0A A4 4C 0A 88 D0 FC 90
0A70: 08 A9 FE 25 01 85 01 B0 1F C8 C8 A5 4A D1 0E F0
0A80: 04 B0 0A 90 13 A5 49 C8 D1 0E 90 0C 88 C8 A9 01
0A90: 45 01 85 01 C0 09 D0 E2 06 01 CA 10 BF 66 01 E8
0AA0: E0 14 F0 25 20 D5 0A 98 C5 4C 90 F3 D0 1B B5 2A
0AB0: 4A 4A C5 4A F0 04 90 0E B0 0F B5 2A 29 03 A8 A5
0AC0: 49 D9 F3 0F 90 03 4C 9F 0A 8A 29 01 0A 0A 05
0AD0: 01 8D 00 04 60 8A 48 4A AA B5 3E 90 08 4A 4A 4A
0AE0: 4A A8 68 AA 60 29 0F 10 F8 A2 00 8A 09 60 85 82
0AF0: 8A C9 09 D0 02 E8 8A 4A A8 90 09 B9 48 00 4A 4A
0B00: 4A 4A 10 03 B9 48 00 29 0F 09 30 85 82 E8 E0 0E
0B10: D0 D9 60 A9 40 85 1B A2 04 A9 02 85 1C B5 65 C6
0B20: 1C F0 27 30 29 4A 4A 4A 4A A8 B9 DB 0F A4 1B 84
0B30: 82 85 80 20 86 0B A5 21 C9 01 F0 07 25 1B F0 03
0B40: 20 86 0B 88 84 80 46 1B 10 D3 29 0F 10 DB CA F0
0B50: 09 E0 01 D0 C4 B5 65 4C 2D 0B A5 21 F0 04 C9 01
0B60: D0 03 20 86 0B A9 00 85 82 A9 BF 85 80 A5 82 D0
0B70: 09 A5 80 09 80 6A 85 80 B0 F3 A5 80 09 80 85 62
0B80: A2 FF 86 80 AA 60 A0 64 88 D0 FD 60 20 3C 09 C9
0B90: FD D0 F8 20 13 0B A2 7F 9A 20 44 0E 20 13 0B E8
0BA0: D0 FA 20 13 0B E8 F0 FA 20 13 0B E8 F0 F4 C9 EF
0BB0: F0 11 C9 BF F0 13 C9 DF F0 06 C9 FD F0 47 D0 E2
0BC0: 4C AF 0C 20 54 0E 4C 9C 0B A2 FF 86 69 86 68 A9
0BD0: 2A C5 0A F0 1D A9 02 C5 0B D0 02 86 66 A9 0A C5
0BE0: 0B F0 09 20 E6 0E E6 0B E6 0B 10 F1 20 48 0E 4C
0BF0: 9C 0B 86 66 20 E6 0E E6 0B A9 14 C5 0B D0 F5 20
    
```

Reihe geschaltete 4,5-V-Flachbatterien, dann sollten sie alle 1 bis 2 Jahre ausgetauscht werden. Im Falle der Batterieversorgung entfällt R35. Hält man eine Notstromversorgung ganz und gar für überflüssig, dann entfallen die Akkus, R35 und D9. Anstelle von D8 wird in die Platine eine Drahtbrücke gesetzt. Die Adressenkodierung ist mit nur zwei Invertiern sehr einfach ausgeführt, da der gesamte Speicherbereich in nur drei Blöcke aufgeteilt wird (IC2, IC3 und IC4). Der Prozessor kann zwar einen Bereich von 64 K adressieren. In diesem Fall "wiederholt sich" der Prozessor aber in 4 K Speicherblöcken. Auf die Arbeitsweise hat das allerdings keinen nachteiligen Einfluß. Die drei Blöcke

werden mittels der Adressenleitung A10 und A11 dekodiert:

A11	A10	
0	0	IC2
0	0	IC4
1	0	
1	1	IC3

Der Speicherbereich sieht dann folgendermaßen aus:

*000	*400	*800
.	.	.
.	.	.
.	IC2	IC4
.	.	IC3
.	.	.
.	.	.
*3FF	*7FF	*FFF

(* = "dont care")

```

OC00: 44 OE 4C 9C 0B 20 44 OE A9 02 C5 0B F0 25 A5 69
OC10: 85 1A A5 68 85 1B C6 0B C6 0B 20 7C 0E A5 69 C5
OC20: 1A F0 04 B0 5F 90 08 A5 68 C5 1B F0 02 B0 55 E6
OC30: 0B E6 0B A9 02 C5 0A D0 06 A9 08 C5 0B F0 06 20
OC40: 54 OE 4C 08 0C 20 48 OE A5 69 85 5B A5 68 85 5C
OC50: 20 D5 0A C0 08 F0 28 84 1A E8 86 0B 20 AF 0E 98
OC60: C5 1A 90 20 F0 02 B0 11 EA A5 69 C5 5B 90 15 F0
OC70: 02 B0 06 A5 68 C5 5C 90 0B A5 0B C9 13 D0 C9 C6
OC80: 1D 4C 37 08 A9 CD 85 69 A9 DA 85 68 A9 DF 85 67
OC90: A9 FF 85 66 85 0D 20 13 0B C6 0D D0 F9 A9 2A C5
OCA0: 0A F0 06 20 50 0E 4C 9C 0B 20 68 0E 4C 9C 0B A6
OCB0: 69 E8 D0 04 86 69 86 68 A9 80 85 21 A9 2A C5 0A
OCC0: F0 30 A9 02 C5 0B F0 2A 46 21 A9 0C C5 21 D0 04
OCD0: A9 04 85 21 A9 02 C5 21 F0 1B A9 40 C5 21 F0 1C
OCE0: 4A C5 21 F0 30 4A C5 21 F0 25 4A C5 21 F0 23 4C
OCF0: 83 0D 4C 96 0D A9 00 85 21 4C 9C 0B 20 4D 0F A5
OD00: 69 18 69 10 C9 24 90 02 A9 00 85 69 4C FC 0C 4C
OD10: 3C 0D 4C 6E 0D 20 4D 0F E6 69 A9 20 25 69 F0 0C
OD20: A5 69 C9 24 D0 EF A9 20 85 69 10 E9 A5 69 29 0F
OD30: C9 0A D0 E1 A9 F0 25 69 85 69 10 D9 A9 2A C5 0A
OD40: F0 13 20 4D 0F A5 68 18 69 10 C9 60 90 02 A9 00
OD50: 85 68 4C 42 0D A9 18 85 21 20 4D 0F 18 A5 68 F8
OD60: 69 15 D8 C9 60 D0 02 A9 00 85 68 4C 59 0D 20 4D
OD70: 0F E6 68 A5 68 29 0F C9 0A D0 F3 A9 F0 25 68 85
OD80: 68 10 EB 20 4D 0F A2 2A E4 0A D0 04 A2 FF 86 0A
OD90: 20 48 0E 4C 83 0D A9 00 85 21 20 97 0F E8 D0 FA
ODA0: 20 97 0F E8 F0 FA 20 97 0F E8 F0 F4 C9 DF F0 49
ODB0: C9 EF F0 4F C9 FB D0 E8 A9 2A C5 0A F0 4B A9 80
ODC0: 85 0C A9 FF 85 66 46 0C 20 B6 0F E8 D0 FA 20 B6
ODD0: 0F E8 F0 FA 20 B6 0F E8 F0 F4 C9 DF F0 1B C9 EF
ODE0: F0 21 C9 FB F0 0A C9 F7 D0 E4 A5 0C 45 66 85 66
ODF0: A9 01 C5 0C D0 D0 4C 9A 0D 20 E6 0E A9 80 85 21
OE00: 4C C8 0C 20 E6 0E 4C C3 0B A0 01 A6 0B F0 04 CA
OE10: 20 D5 0A B9 EA 0F 85 66 49 FF 85 0C 20 B6 0F E8
OE20: D0 FA 20 B6 0F E8 F0 FA 20 B6 0F E8 F0 F4 C9 FB
OE30: F0 07 C9 F7 D0 EC 4C 96 0D 46 0C A5 0C F0 CA 38
OE40: 66 66 30 D8 A9 FF 85 0A E6 0A A9 03 C5 0A F0 18
OE50: A9 00 85 0B A9 2A C5 0A F0 16 E6 0B E6 0B A9 0A
OE60: C5 0B F0 E4 20 7C 0E 60 A9 2A 85 0A A9 FF 85 0B
OE70: E6 0B A9 14 C5 0B F0 CC 20 AF 0E 60 A9 00 A8 85
OE80: 1D A5 0A 0A 0A 0A 0A AA 05 0B 85 1C B1 1C 85 69
OE90: E6 1C B1 1C 85 68 86 1C A5 0B 6A 6A 90 0D A9 0B
OEA0: 05 1C 85 67 86 1C B1 1C 85 66 60 A9 0A 10 F1 A5
OEB0: 0B AA B5 2A A8 C8 F0 23 29 FC 4A 4A 85 69 B5 2A
OEC0: 29 03 A8 B9 F3 0F 85 68 8A 4A 90 16 A9 0A 09 30
OED0: 85 67 20 D5 0A B9 EA 0F 85 66 60 88 84 69 84 68
OEE0: 30 E6 A9 0B 10 E8 A5 0A C9 2A F0 18 0A 0A 0A 0A
OEF0: 85 1C A0 00 A5 66 91 1C A4 0B A5 69 91 1C C8 A5
OF00: 68 91 1C 60 A0 00 A5 0B AA A5 69 0A 0A 95 2A A5
OF10: 68 F0 0B C8 C9 15 F0 06 C8 C9 30 F0 01 C8 98 15
OF20: 2A 95 2A A0 00 A9 7F C8 38 6A C5 66 D0 F9 8A 4A
OF30: AA B5 3E 90 0E 29 0F 95 3E 98 0A 0A 0A 0A 15 3E
OF40: 95 3E 60 29 F0 95 3E 98 15 3E 95 3E 60 20 81 0F
OF50: E8 D0 FA 20 81 0F E8 F0 FA 20 81 0F E8 F0 F4 C9
OF60: DF F0 09 C9 EF F0 0F C9 FB D0 E8 60 BA E8 E8 9A
OF70: 20 E6 0E 4C C8 0C A9 00 85 21 BA E8 E8 9A 4C 03
OF80: 0E A5 21 85 1A E6 0D E6 0D 10 04 A9 00 85 21 20
OF90: 13 0B A4 1A 84 21 60 E6 0D A9 70 25 0D D0 12 A5
OFA0: 66 49 FF 85 66 20 13 0B A5 66 49 FF 85 66 A5 62
OFB0: 60 20 13 0B D0 F8 E6 0D E6 0D 10 14 A5 66 85 1A
OFC0: A5 0C 45 66 85 66 20 13 0B A5 1A 85 66 A5 62 60
OFD0: 20 13 0B D0 F8 18 90 FE FF FF FF 40 79 24 30 19
OFE0: 12 02 78 00 10 23 7B 06 2F FF FF BF DF EF F7 FB
OFF0: FD FE FF 00 15 30 45 FF FF FF D5 0F 00 08 16 0A
    
```

recht ausführlich beschrieben. Wie die Uhr funktioniert, wurde noch nicht erklärt. Das ist bei einem Programm in Maschinensprache gar nicht so einfach. Um das Ganze nicht unnötig kompliziert zu machen, wird im folgenden nur das Nötigste an Software beschrieben.

Über PA7 kommt die Zeitinformation von DCF77 an. Die Nullen und Einsen werden aus der Länge der Impulse und deren Wertigkeit dekodiert. Außerdem wird der Beginn jeder Minute detektiert und "behalten", wieviel Impulse in einer Minute empfangen wurden. Ist die richtige Menge von Impulsen eingetroffen, wird die empfangene Information gespeichert. Sonst beginnt der Prozessor aufs neue zu zählen und zu dekodieren. Ist auch die Information der darauf folgenden Minute eingetroffen, werden beide Datenmengen miteinander verglichen. Unterscheiden sich beide nur dadurch, daß sie eine Minute auseinander liegen, wird die Information der letzten empfangenen Minute auf dem Display angezeigt. Die Sekunden-Information ist darin allerdings nicht enthalten. Sie stammt aus dem Timer in IC2, der aus der Quarzfrequenz die Sekunden-Impulse erzeugt. Solange die Anzeige auf dem Display nicht vom DCF-Sender stammt, die Uhr also mit dem internen Quarzoszillator läuft, blinkt die "Wochentag-LED". Das Uhr-Programm sorgt außerdem dafür, daß die LEDs und das Display gemultiplext und die Tasten S1...S7 abgefragt werden. Gleichzeitig erfolgt natürlich die Speicherung aller Schaltzeiten, wird kontrolliert, ob diese Zeiten in der richtigen Reihenfolge eingegeben wurden und daß einer der Ausgänge T0...T3 zu der auf dem Display angezeigten Zeit ein- oder ausgeschaltet wird. Das Programm für die Uhr ist über den Elektor-Software-Service erhältlich. Wer das Programm gerne selbst in ein EPROM "schießen" möchte, (eine Heidenarbeit!), findet es in der Hex-Dump-Darstellung in Tabelle 1.

Aufbauen und Testen

In den Bildern 2 und 3 sind die Platinen dargestellt. Eine Platine nimmt die Siebensegmentanzeigen, die LEDs und die Tasten auf. Die andere Platine enthält den Prozessor mit allen anderen Bauteilen einschließlich der Versorgung. Die Platinen sind in ihren Abmessungen so gewählt, daß man sie mit den Lötseiten gegeneinander montieren kann. Beim Einbau in ein Gehäuse muß man allerdings bei dieser Sandwich-Konstruktion aufpassen. Einige Befestigungslöcher befinden sich inmitten breiter Leiterbahnen. Hier darf man nur isolierendes Befestigungsmaterial verwenden, sonst können Kurzschlüsse entstehen! Für die "Wochentag-LEDs" nimmt man am besten flache, rechteckige Typen (z. B. HP 5082-4670). Mit Abreibuchstaben können die Wochentage dann auf die LEDs gerubbelt werden. Man kann auch "normale" LEDs ver-

Die gewählte Reihenfolge ist nicht zufällig. Das EPROM liegt oben im Speicherbereich, weil von dort die NMI-, RESET- und IRQ-Vektoren geholt werden müssen. Das RIOT-IC (RIOT = RAM, I/O, TIMER) ist aus zwei Gründen ganz unten in den Speicherbereich gelegt worden: Mit dem 6502-Prozessor kann man bestimmte "Seite Null"-Befehle ausführen (Seite Null umfaßt die Adressen 0000...00FF), die nur 2 Byte lang sind. Werden diese Befehle in anderen Seiten gebraucht, sind sie drei Byte lang. Man erzielt damit also eine erhebliche Programmeinsparung. Seite 1 (0100...01FF) muß einen RAM-Speicher enthalten, der als Stack arbeitet. Das ist dadurch realisiert, daß

die Adressenleitungen A8 und A9 nicht an IC2 angeschlossen sind (das RIOT-IC liegt dann nämlich auf den Seiten 0, 1, 2 und 3).

Der 128-Byte-RAM-Speicher in IC2 dient verschiedenen Zwecken. Der untere Bereich wird in Seite Null (0000...0069) als Informations-"Lager" für Zwischen- und Schaltzeiten verwendet. Der restliche Teil von Seite 1 (016A...017F) wird als Stack gebraucht. Schließlich verwendet man den Adressbereich zwischen RIOT und EPROM für den Zwischenspeicher IC4.

Nur ein bißchen Software

Wir haben zwar nun die Schaltung

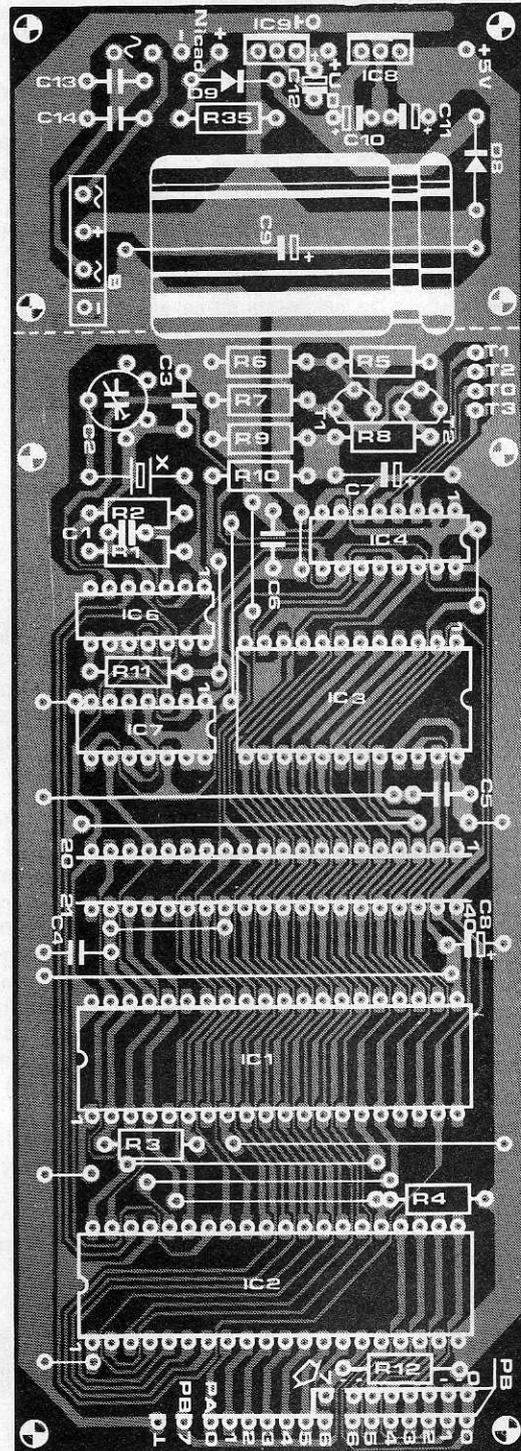
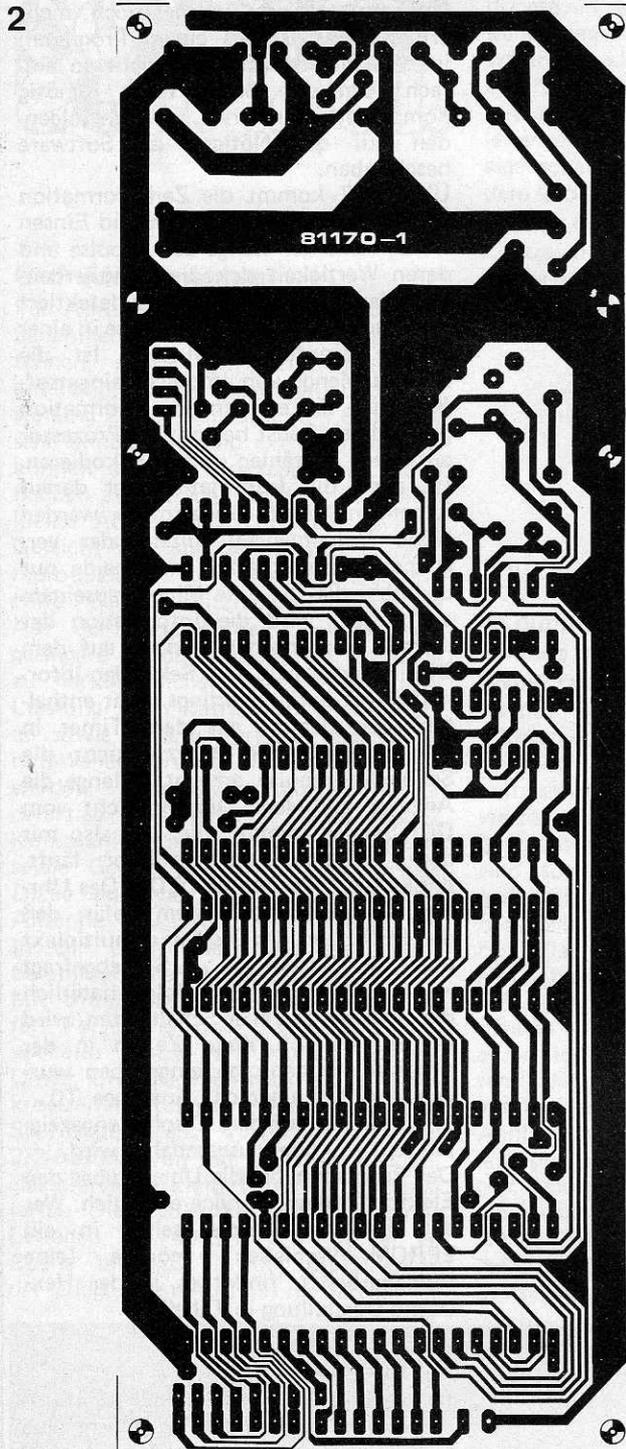


Bild 2. Hauptplatine mit Bestückungsplan. Die Platine kann aufgrund des universellen Konzepts auch für andere Projekte verwendet werden. Den Versorgungs-Teil der Platine kann man bei Bedarf auch abtrennen.

Stückliste

Widerstände:

R1, R2, R7 = 2k2
 R3, R4, R12 = 3k3
 R5 = 1 k
 R6 = 5k6
 R8 = 56 Ω
 R9 = 560 Ω
 R10 = 470 Ω
 R11 = 15 k
 R13 = 220 Ω
 R14 ... R20 = 12 k
 R21 ... R27 = 10 k
 R28 ... R34 = 10 Ω
 R35 = 120 Ω

Kondensatoren:

C1 = 10 n keramisch
 C2 = 4 ... 40-pF-Trimmer
 C3 = 150 p keramisch
 C4, C5, C6, C13, C14 = 100 n
 C7 = 47 μ /6,3 V
 C8, C11, C12 = 10 μ /10 V Tantal
 C9 = 2200 μ /25 V
 C10 = 10 μ /16 V Tantal

Halbleiter:

B = B40C1500 Brückengleich-
 richter
 D1 ... D7 = LED rot (siehe Text)
 D8, D9 = 1N4001
 LD1 ... LD6 = DL 7760
 (gem. Kathode)
 T1, T2 = BC 557
 T3 ... T9 = BC 516
 IC1 = 6502
 (Synertec, Rockwell)

IC2 = 6532

(Synertec, Rockwell)
 IC3 = 2716 (keine Texas
 Instruments 2716!)
 IC4 = 74LS173
 IC5 = ULN 2003 oder XR 2003
 IC6 = 74LS04
 IC7 = 74LS74
 IC8, IC9 = 7805

außerdem:

Tr = Netztrafo 10 V/1,5 A sek.
 S1 ... S7 = Digitast
 X1 = 4-MHz-Quarz
 2 x Kühlkörper für 7805

3

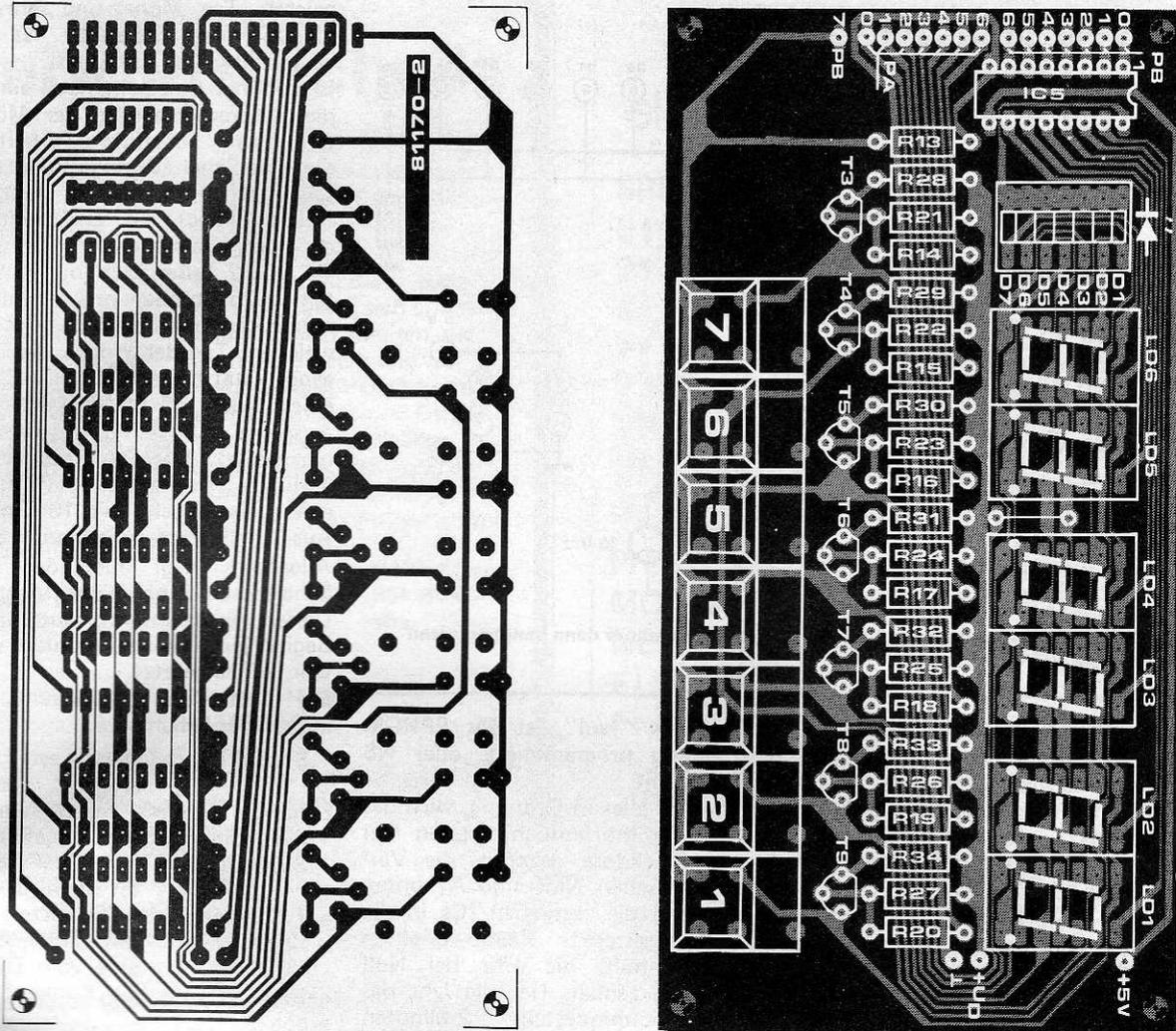


Bild 3. Displayplatine mit Bestückungsplan. Die Platine enthält Siebensegmentanzeigen, "Wochentag-LEDs", die Tasten und die Steuerelektronik.

wenden und die Wochentage daneben auf die Frontplatte schreiben. Schließlich kann auch ein sogenanntes LED-Array aus 10 LEDs (z.B. MV 57164) eingesetzt werden, dann muß man allerdings 3 LEDs vorsichtig (!) abtrennen oder abdecken.

Die beiden Stabilisator-ICs müssen gut gekühlt werden. Ist das Uhren-Gehäuse aus Metall, kann man es gleichzeitig als Kühlkörper verwenden. Die ICs werden direkt mit reichlich Wärmeleitpaste auf das Gehäuse gesetzt. Allerdings darf man in diesem Fall keinen anderen Punkt der Schaltung mit dem Gehäuse verbinden. Diese Konstruktion sollte übrigens so ausgeführt sein, daß die ICs direkt in die Platine eingelötet werden können. Lange Drahtverbindungen vermeiden! Man kann den Versorgungsteil der Platine auch abtrennen und irgendwo anders im Gehäuse befestigen. Die Platinen werden so gegeneinander gesetzt, daß die Anschlüsse PB0...PB6, PA0...PA6 und PB7 sich genau gegenüber liegen. Die Verbindung untereinander geschieht dann mit kurzen Drahtstücken. Schließlich werden dann noch die drei Betriebsspannungsan-

schlüsse jeder Platine miteinander verbunden.

Uhr und Normalzeitempfänger können in ein Gehäuse eingebaut werden, wenn man darauf achtet, daß die Platinen mit passenden Blechen oder Printmaterial gegeneinander abgeschirmt sind. Das oder die Abschirmblech(e) sind mit Masse zu verbinden!

Die Aktive Antenne wird über ein Koaxkabel und eine passende (BNC-) Buchse an den Empfänger angeschlossen. Ist die Uhr vollständig aufgebaut, kann man alle ICs in die Fassungen stecken, den Transformator und das Netz anschließen und abwarten, ob die Uhr irgendwelche Lebenszeichen von sich gibt. Nach minimal 2 Minuten muß sie sich auf die richtige Zeit eingestellt haben. Liegt irgendein Fehler vor, dann bereitet die Suche danach ohne komplizierte Meßgeräte schon einiges Kopfzerbrechen. Die einfache Fehlersuche mittels Oszilloskop und Multimeter ist im folgenden beschrieben.

Zunächst werden alle ICs wieder aus den Sockeln gezogen. Nur die Stabilisatoren IC8 und IC9 bleiben angeschlossen. Auch die Akkus oder Batterien werden

entfernt. Man mißt dann, ob die Ausgänge der Stabilisator-ICs tatsächlich 5 V liefern. Ist das der Fall, schaltet man die Uhr ab und setzt IC6 und IC7 ein. Uhr wieder einschalten und mit dem Oszilloskop überprüfen, ob an Pin 8 von IC7 ein symmetrisches 1-MHz-Taktsignal liegt. Ohne Oszilloskop geht's auch – mit einem Multimeter und der Hilfsschaltung Bild 4a. Das Instrument muß ungefähr 0 V anzeigen. Ist das der Fall, arbeitet der Oszillator jedenfalls. Eine genaue Frequenzeinstellung ist mit dem Zähler möglich. Den Oszillator mit C2 abgleichen. Dabei läßt sich eine maximale Ganggenauigkeit von 0,5 s pro Tag erreichen.

Im nächsten Schritt wird nachgesehen, ob das RES-Signal an den Pins 9 und 10 von IC6 logisch 1 ist. Ist auch das in Ordnung, setzt man mittels einiger Drahtstücke, wie in Bild 4b dargestellt, den Kode AA auf den Datenbus. Die angegebenen Zahlen beziehen sich auf die Anschlüsse der Verbindungsleiste zwischen IC1 und IC3 auf der Platine.

Klappt bis hierhin alles, wird die Uhr ausgeschaltet und IC1, der Prozessor, in die Fassung gesetzt. Nach dem Ein-

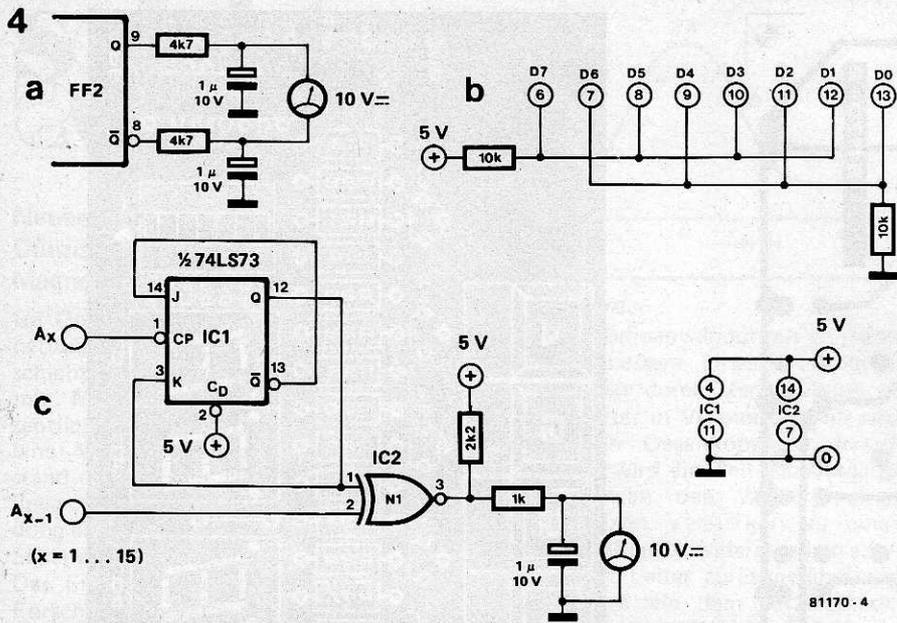


Bild 4. Drei Hilfsschaltungen zur Überprüfung der Uhr. Sie werden nur dann benötigt, wenn kein Oszilloskop zur Verfügung steht.

schalten muß auf A0 (Verbindungsleiste Pin 29) eine symmetrische Rechteckspannung mit einer Frequenz von 250 kHz liegen. Entsprechendes gilt für A1 (f = 125 kHz), A2 (f = 62,5 kHz) und so weiter bis A15 (f = 7,6 Hz). Außerdem muß das R/W-Signal (Verbindungsleiste Pin 14) ständig logisch 1 sein. Verläuft eine dieser Prüfungen nicht zufriedenstellend, kann man noch einmal nachschauen, ob auf dem Datenbus wirklich "AA" liegt. Auch diese Messung kann mit Hilfe der Schaltung in Bild 4c ohne Oszilloskop durchgeführt werden. Die Schaltung wird an jeweils benachbarte Adressenleitungen beginnend mit dem Paar A15/A14 angeschlossen. Bei jeder Messung muß das Instrument ständig 0 oder 5 V anzeigen. Zeigt es irgendeinen Zwischenwert an, dann gibt es zwischen den Bahnen entweder einen Kurzschluß, oder eine Bahn ist unterbrochen. Am besten kontrolliert man zunächst, ob auf A15 das Signal mit der Frequenz 7,6 Hz liegt. Die Instrumentennadel wird bei dieser niedrigen Frequenz sichtbar vibrieren. Danach führt man die Messungen mit der erwähnten Hilfsschaltung durch. Ist alles in Ordnung, wird der Kode AA wieder vom Datenbus genommen. Daran denken, daß nur ohne IC1 in der Platine gelötet wird! Die Versorgung wurde vorher ebenfalls abgeschaltet – oder? Das EPROM wird eingesetzt und Punkt 26 der Verbindungsleiste (NMI) mit Punkt 36 (A7) verbunden. Nach dem Einschalten der Betriebsspannung müssen die Adressenleitungen folgende logische Zustände haben:

A15 A14 A13 A12 A11 A10
0 0 0 0 1 1

A9 A8 A7 A6 A5 A4 A3... A0

1 1 1 1 0 1 nicht stabil
Außerdem muß an Pin 20 von IC3 ständig eine logische 0 liegen. Ist hier

irgend etwas "faul", ist das EPROM nicht richtig programmiert, oder N5 invertiert nicht.

Ist allerdings alles in Ordnung, wird der Stecker zum (hoffentlich) letzten Mal aus der Steckdose gezogen, die Verbindung zwischen NMI und A7 unterbrochen und die restlichen ICs in die Fassungen gesteckt. Beim erneuten Einschalten muß die Uhr bei Null anfangen zu zählen. Hat die Uhr das DCF-Signal mindestens 2 Minuten lang empfangen, muß sie sich selbst stellen.

Noch ein Wort zum "genauen" Abgleich des Quarz-Oszillators mit Hilfe eines Frequenzzählers. Wir erwähnten schon, daß solch ein Abgleich mittels C2 möglich ist, wenn an Pin 8 von IC7 ein Frequenzmesser angeschlossen wird. Allerdings gibt es im Hobby-Labor wohl kaum einen wirklich genauen Frequenzzähler. Es gibt aber auch eine andere, einfache Abgleichmethode: Der Trimmer C2 wird zunächst in Mittelposition gesetzt. Man läßt die Uhr sich stellen und trennt danach den Empfänger ab. Ist die Uhr einige Stunden mit der eigenen Zeitbasis gelaufen, schließt man den Empfänger wieder an. Synchronisiert die Uhr nach etwa 2 Minuten, dann sieht man, ob der Oszillator zu schnell oder zu langsam läuft. Mit C2 kann man nachstellen. Wird diese allerdings sehr zeitraubende Methode mehrfach wiederholt, kann man schließlich sicher sein, einen sehr genau abgeglichenen Oszillator zu haben.

Die Schaltuhr

Die Bedienung der "normalen" Uhr bereitet keine Schwierigkeiten. Sie stellt sich selbst, und das Display zeigt Stunden, Minuten, Sekunden und den Wochentag mittels einer LED an. Beim Betätigen der DATE-Taste S1 erscheinen

auf dem Display, von links nach rechts gelesen, Tag, Monat und Jahr. Die Bedienung der Schaltuhr ist dagegen vergleichsweise kompliziert. Vor der Bedienungs-Anleitung noch einmal kurz die Zusammenfassung der Möglichkeiten, mit der Schaltuhr zu arbeiten: Über die vier Schaltausgänge kann man mittels mechanischer oder elektronischer Relais (Triac) ein elektrisches Gerät ein- und ausschalten. Die Ausgänge T0...T2 lassen sich mit 4 Schaltzeiten in 25 Stunden programmieren. Daneben kann man auch noch angeben, an welchem Tag der Woche die Schaltvorgänge stattfinden sollen. Zu Beginn jedes Tages, um 00.00 Uhr, werden die Ausgänge T0...T2 automatisch zurückgesetzt. Der Schaltabstand beträgt 1 Minute. Der Ausgang T3 kann über eine ganze Woche mit 10 Ein- und 10 Aus-Schaltzeiten programmiert werden. Allerdings liegt hier der minimale Schaltabstand bei einer Viertelstunde. Dieser Ausgang wird automatisch zu Beginn jeder Woche, montags um 00.00 Uhr, zurückgesetzt.

Die Tastenfunktionen zum Programmieren der Schaltzeiten:

- S1, DATE Diese Taste dient, wie erwähnt, nur zur Anzeige des Datums, Sie ist hier nur der Vollständigkeit halber aufgeführt.
- S2, MODE Mit dieser Taste schaltet man die Anzeige auf "Uhrzeit" oder "Schaltzeit".
- S3, > Mit dieser Taste wird eine vom Display angezeigte Ziffer (die blinkende) um 1 erhöht.
- S4, SET DAY Diese Taste dient nur zum Programmieren des Wochentags.
- S5, NEXT Nach Betätigen dieser Taste erscheint ständig die nächstfolgende Schaltzeit auf dem Display.
- S6, CURSOR Mit dieser Taste kann der Cursor von links nach rechts über das Display geschoben werden (Achtung: Die Siebensegmentanzeige ganz rechts zeigt an, ob die Schaltzeit für Ein- oder Ausschalten gilt.). Die vom Cursor angesteuerte Siebensegmentanzeige blinkt zum Zeichen dafür, daß sie mit Hilfe der >-Taste verändert werden kann.
- S7, CLEAR Hiermit können alle oder einige der Schaltzeiten auf einer Leitung (Ausgang) gelöscht werden. Der Löschvorgang beginnt mit der aktuellen Display-Anzeige.

Die Siebensegmentanzeige ganz rechts gibt an, ob die angezeigte Schaltzeit für Ein- oder Ausschalten gilt. "1" bedeutet Einschaltzeit, und "0" bedeutet Ausschaltzeit. Die folgende Siebensegmentanzeige gibt den Schaltausgang an: 0, 1, 2 oder 3. Wir geben auf den Seiten 9-66 und 9-67 ein Beispiel an. Dort kann man anhand der Kommentare gut erkennen, wie programmiert wird, und welche Möglichkeiten man nutzen kann.

Tabelle 2. Daten-Ausgabe im ASCII-Kode

	PB6	PB5	PB4	PB3	PB2	PB1	PB0		PB6	PB5	PB4	PB3	PB2	PB1	PB0
A	1	1	0	0	0	0	0	B	0	1	1	*	*	*	*
C	1	1	0	0	0	0	1	D	0	1	1	*	*	*	*
E	1	1	0	0	0	0	1	F	0	1	1	*	*	*	*
G	1	1	0	0	0	1	1	H	0	1	1	*	*	*	*
I	1	1	0	0	1	0	0	J	0	1	1	*	*	*	*
K	1	1	0	0	1	0	1	L	0	1	1	*	*	*	*
M	1	1	0	0	1	1	0	N	0	1	1	*	*	*	*
O	1	1	0	0	1	1	1	P	0	1	1	*	*	*	*
Q	1	1	0	1	0	0	0	R	0	1	1	*	*	*	*
S	1	1	0	1	0	0	1	T	0	1	1	*	*	*	*
U	1	1	0	1	0	1	0	V	0	1	1	*	*	*	*
W	1	1	0	1	0	1	1	X	0	1	1	*	*	*	*
Y	1	1	0	1	1	0	0	Z	0	1	1	*	*	*	*

in einem Computersystem die richtige Zeit auszuwerten oder anzuzeigen. Die Information (Zeit und Datum) wird ein Mal pro Sekunde während 600 µs ausgegeben. Man kann leicht zwischen Daten-Ausgabe und Display-Routine unterscheiden. Während des Ablaufs der Display-Routine ist immer höchstens eine der Leitungen PB6, PB5 und PB4 logisch 1. Während der Daten-Ausgabe sind immer zwei dieser Leitungen gleichzeitig "1" (PB4 und PB5 oder PB5 und PB6). Der Kode bei Daten-Ausgabe ist folgendermaßen aufgebaut: Erst erscheint

PB6	PB5	PB4	PB3	PB2	PB1	PB0
1	1	0	*	*	*	*

Darin steckt die Information, daß direkt, nachdem PB6 "0" geworden ist, die Daten-Ausgabe beginnt. Die Information auf PB0...PB3 (****) gibt an, um welche Daten es sich handelt. Darauf folgt 001****, der ASCII-Kode. In Tabelle 2 ist angegeben, in welcher Reihenfolge die Daten-Ausgabe stattfindet. Zur Verdeutlichung ist in Bild 5 ein Foto des Impulsdigramms der Zeit- und Datum-Ausgabe dargestellt. Die dort angegebenen Bezeichnungen stimmen mit denen der Tabelle 2 überein.

5

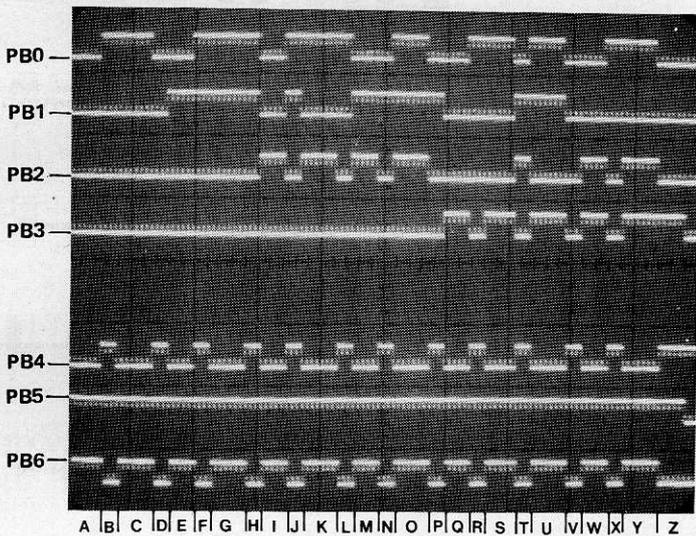


Bild 5. Das Impulsdigramm der ASCII-Daten-Ausgabe. Die Buchstaben beziehen sich auf die Übersicht in Tabelle 2. Zeit und Datum sind hier: 13.33.01 Uhr und 22-06-81 Montag (1).

Schließlich ...

...genug der Beschreibung dieser "Super-de-Luxe-Schaltuhr". Sicher wird mancher mit den Erklärungen der Mikroprozessor-Technik seine Schwierigkeiten haben. Aber der Artikel ist ja auch kein grundlegender über Mikroprozessoren. Bei der DCF-Schaltuhr ist der µP "nur" Mittel zum Zweck, für den Nachbau allein braucht man sich mit dieser Technik nicht weiter auseinanderzusetzen. Da müßte sich der interessierte Leser schon etwas genauer mit dem Junior-Computer beschäftigen. Aber etwas ist doch dran an der gedanklichen Verbindung: Computer-Schaltuhr - Mikroprozessorsystem. Inmitten der größeren Platine der Uhr gibt es nämlich eine ganze Menge von Anschlüssen, die gar nicht benötigt werden. Die Platine ist allerdings in dieser Ausführung für eine Art von Mini-Basis-Mikroprozessorsystem zu gebrauchen, das in Zukunft, mit eben dieser Platine, für andere Projekte entwickelt werden kann. Aber das ist noch Zukunftsmusik!

Kehrt man nach dem Betätigen der MODE-Taste wieder in das Uhrzeit-Programm zurück, dann werden die Ausgänge T0...T3 an die programmierten Schaltzeiten angepaßt. Das geschieht genau eine Sekunde, nachdem eine Minute vergangen ist. Während der Schaltzeit-Programmierung bleibt der Zustand der Ausgänge unverändert. Nach der Rückkehr ins Uhrzeit-Programm fällt außerdem auf, daß die "Wochentag-LED" blinkt. Das rührt daher, daß während der Schaltzeit-Programmierung das Zeitsignal nicht dekodiert wird. Nach mindestens 2 Minuten stellt die Uhr sich dann wieder auf die richtige Zeit ein. Ein Punkt wurde noch nicht erwähnt. Wird eine Ausschaltzeit so program-

miert, daß sie vor einer Einschaltzeit liegt, dann erscheint beim Betätigen der MODE-Taste einige Zeit lang "Error" (Fehler) auf dem Display. Danach verlischt der "Error"-Schriftzug; und die erste Schaltzeit des Ausgangs, in dem der Fehler steckt, wird eingeblendet. In diesem Fall kann man nicht in das Uhrzeit-Programm zurückkehren. Erst wenn der Fehler korrigiert ist, erscheint nach dem Betätigen der MODE-Taste die Uhrzeit auf dem Display.

ASCII-Daten-Ausgabe

Auf der größeren Platine sind die Punkte PB0...PB6 extra an eine Seite geführt. Dort sind Zeit und Datum im ASCII-Kode verfügbar. Man kann diese Information beispielsweise dazu nutzen,

Programmierbeispiel

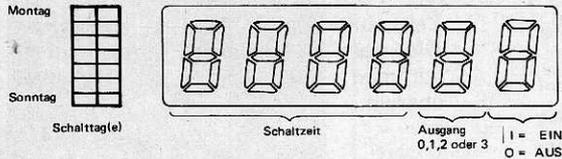
Ausgang T0: Montag und Freitag
um 8.30 h ein- und
um 9.02 h ausschalten

Ausgang T1: immer "0"

Ausgang T2: immer "0"

Ausgang T3: Sonntag 20.00 h einschalten
Dienstag 8.00 h ausschalten
Mittwoch 10.00 h einschalten
Donnerstag 00.45 h ausschalten

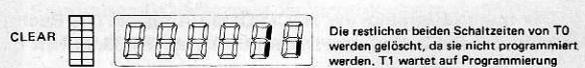
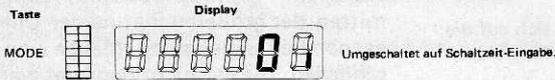
Anzeige der Schaltzeit auf dem Display:



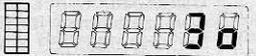
Programmierung:

Erklärung der Symbole:

- Wochentag-LED leuchtet nicht
- Wochentag-LED leuchtet
- Wochentag-LED blinkt
- Siebensegmentanzeige blinkt
- Siebensegmentanzeige leuchtet
- Siebensegmentanzeige leuchtet nicht



SET DAY  Montag programmiert. Da T3 immer in der Nacht von Sonntag auf Montag um 00.00 h auf "0" gesetzt wird, muß die erste Einschaltzeit Montag 00.00 h sein

NEXT  Die nächste (Aus-)Schaltzeit soll programmiert werden.

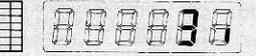
CURSOR  "Wochentag-LEDs" blinken, warten auf Programmierung

> (2 mal)  "Dienstag-LED" blinkt

SET DAY  Dienstag programmiert

ISOR (2 mal)  Stunden-Einer blinken, warten auf Programmierung

> (8 mal)  Stunden-Einer programmiert

NEXT  Die nächste (Ein-)Schaltzeit soll programmiert werden

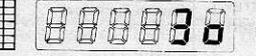
CURSOR  "Wochentag-LEDs" blinken, warten auf Programmierung

> (2 mal)  Nach der ersten Betätigung dieser Taste blinkt die "Dienstag-LED". Die zu programmierende Schaltzeit muß auf Dienstag oder später fallen

SET DAY  Mittwoch programmiert

CURSOR  Stunden-Zehner blinken, warten auf Programmierung

>  Stunden-Zehner programmiert

NEXT  Die nächste Schaltzeit soll programmiert werden

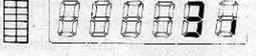
CURSOR  "Wochentag-LEDs" blinken, warten auf Programmierung

> (2 mal)  "Donnerstag-LED" blinkt

SET DAY  Donnerstag programmiert

CURSQR (3 mal)  Minuten-Einer und -Zehner blinken, da die Schaltzeiten bei T3 im Viertelstunden-Abstand liegen

> (3 mal)  Jede Betätigung dieser Taste erhöht die Anzeige um 15 Minuten. Minuten programmiert

NEXT  Die nächste (Ein-)Schaltzeit soll programmiert werden

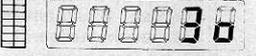
CURSOR  "Wochentag-LEDs" blinken, warten auf Programmierung

> (4 mal)  "Sonntag-LED" blinkt

SET DAY  Sonntag programmiert

CURSOR  Stunden-Zehner blinken, warten auf Programmierung

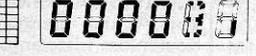
> (2 mal)  Stunden-Zehner programmiert

NEXT  Die nächste Schaltzeit soll programmiert werden

CLEAR  Die restlichen Schaltzeiten von T3 werden gelöscht, da sie nicht programmiert werden.

NEXT  Wird diese Taste wiederholt betätigt, kann man alle Schaltzeiten abfragen. Bei "gelöschten Schaltzeiten" sind nur die Nummer des Ausgangs und der Schaltzustand auf dem Display sichtbar.

CURSOR (5 mal)  Wird der Cursor auf die vorletzte Siebensegmentanzeige gesetzt, kann man von Ausgang zu Ausgang schalten

> (3 mal)  Nach jeder Betätigung dieser Taste erscheint die Einschaltzeit jedes Ausgangs auf dem Display. Mit der NEXT-Taste kann man wieder Schritt für Schritt weiterschalten

CURSOR  Der Cursor wird aus dem Bild geschoben, sonst kann man nicht mit der MODE-Taste auf Anzeige der Uhrzeit umschalten.

MODE  Die Uhrzeit wird angezeigt. Die "Wochentag-LED" blinkt, da die Uhr sich erst wieder stellen muß. Nach minimal 2 Minuten ist die Uhrzeit wieder richtig