

das Programm allerdings gelöscht, falls nicht besondere Vorkehrungen getroffen sind, z. B. Batteriestützung der RAM-Speicherschaltkreise. Ein bereits abgespeichertes Programm ist ebenfalls zu löschen (mit dem Kommando NEW), falls ein neues Programm eingegeben werden soll und der noch freie Speicherplatz nicht ausreicht.

Aus diesen Gründen ist es oft wünschenswert, bereits erarbeitete und erprobte BASIC-Programme auch für längere Zeit zu speichern. Hierfür eignen sich sogenannte externe oder periphere Speichereinheiten. Sie sind zusätzlich an einem Mikrocomputer anschließbar, ergänzen den Hauptspeicher und erhöhen damit das Speichervolumen und die Leistungsfähigkeit eines Mikrocomputers um ein Vielfaches. Bei Kleincomputern werden

z. B. Magnetbandgeräte (Kassettenrekorder), Floppy-Disk-Einheiten (Diskettenlaufwerke) oder Lochbandeinheiten eingesetzt. Als Programm- bzw. Datenträger dienen dabei das Magnetband, kreisförmige, magnetisierbare Folien (Disketten) oder das Papierlochband. Eine Dokumentation des erarbeiteten BASIC-Programms kann man durch Ausdrucken mit einem angeschlossenen Drucker vornehmen.

Für den Amateur ist sicher die Verwendung eines Kassettenrekorders am günstigsten, da ein solches Gerät meist vorhanden und die Speicherung der Programme auf handelsüblichen Magnetbandkassetten möglich ist. Je nach Aufzeichnungsverfahren, Blockungsabständen und Länge der einzelnen BASIC-Anweisungen sind pro Kassette etwa zwi-

schen 1000 und 5000 BASIC-Programmzeilen abspeicherbar. Die meisten BASIC-Systeme stellen wenigstens zwei Kommandos (z. B. SAVE und LOAD) zur Verfügung, mit denen ein im Hauptspeicher vorliegendes BASIC-Programm auf einen externen Speicher transportiert und abgespeichert sowie zu beliebiger Zeit von dort wieder zurück in den Hauptspeicher des Mikrocomputers geladen werden kann. Die Kennzeichnung der einzelnen Programme erfolgt durch einen Programmnamen, den sich der Nutzer entsprechend vorgegebenen Regeln frei wählen kann. Mit Hilfe des Namens markiert der Mikrocomputer ein Programm bei der Abspeicherung und findet es später aus dem gesamten Programmbestand wieder heraus.

(Schluß folgt)

Schaltungen mit Mikroprozessor U 880 D

Eine Melodieklingel mit vielen Melodien

S. LEHMANN

Schon seit geraumer Zeit werden in der Amateurelektronikliteratur Schaltungen mit Mikroprozessor-Bauelementen veröffentlicht. Dieser Trend ist zu begrüßen, da damit die Qualität der Amateurgeräte entscheidend verbessert wird. Dem gegenüber steht jedoch die Scheu vor einer Verwendung dieser Technik, bei der die Funktion nicht mehr verdrätet, sondern

programmiert wird. Die Ursachen dafür sind zum Teil darin zu suchen, daß die bisher veröffentlichten Schaltungen meist umfangreich waren (Amateurcomputer, Mikrorechner). Solche Projekte sind als „Mikroprozessor-Erstellungswerk“ schon zu schwierig. In diesem Beitrag soll deshalb eine kleinere Schaltung vorgestellt werden, die durch ein leichter erreichbares

Erfolgserebnis beim Nachbau die „Mikroprozessor-Hemmschwelle“ überwinden helfen soll.

Diese einfache Schaltung (Bild 1) kommt mit nur vier Schaltkreisen aus, ist aber solchen mit Schieberegister und Tongenerator weit überlegen. Möglich wurde das erst durch Verlagerung des Aufwandes von der Hardware (Schaltung) auf die Software (Programm). Letztere bildet zusammen mit der Minimalschaltung die Funktionen „Schieberegister“, „Tonspeicher“ und „Tongenerator“ nach. Da Software nichts kostet (solange man sie nicht kaufen muß), kann man in ihr die gesamte „Intelligenz“ des Gerätes unterbringen.

Eine herkömmliche Melodieklingel benötigte immer einen steuerbaren Tongenerator. Dieser ist aufwendig und schwierig abzugleichen. Ein Mikroprozessor, ein Ausgabe-Flip-Flop und ein paar Bytes Programm können ihn nicht nur ersetzen, sondern sogar übertreffen. Da der Mikroprozessor nun schon einmal da und mit der Erzeugung der Töne bei weitem nicht ausgelastet ist, kann er auch noch die Ausgabe der Melodien und die Abfrage des Klingelknopfes übernehmen. Durch Opfern einiger weiterer Bytes EPROM-Inhalt ist die Klingel auch schon um eine „Sturmklingel-Sperre“ reicher.

In einem 1-Kbit-EPROM kann man etwa 40 einfache Melodien unterbringen. Na-

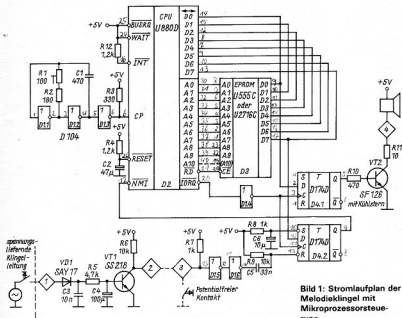


Bild 1: Stromlaufplan der Melodieklingel mit Mikroprozessorsteuerung

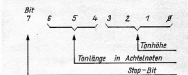


Bild 2: Darstellung des Aufbaus eines Tonbytes

FD 21 3C 01	3E 80 ED 79	76 EA 00 DD	00 D1 00 C5	*.!<.>..yv.....*
00 BA 00 B0	00 A6 00 9C	00 94 00 8B	00 84 00 7C	*.....\
00 75 00 6E	00 68 00 EA	00 A2 00 A9	00 B0 00 B6	*.u.n.h.....*
00 BC 00 C2	00 C8 00 D0	00 D7 00 DD	00 E3 00 B9	*.....*
00 F2 00 FB	00 02 01 4D	00 00 00 04	00 0E 00 ED	*.....M.....*
49 06 4C 10	FE ED 4F 1B	7A B3 C2 4D	00 C3 10 01	*I.L...O.z.M.....*
FF FF FF FF	FF FF FD 7E	00 E6 0F 07	5F 08 7B 08	*.....{
16 00 21 27	00 B7 ED 52	5E 23 56 FD	7E 00 E6 70	*.!'...R^#V...*
0F 0F 0F 0F	47 04 21 00	00 19 10 FD	EB 08 4F 06	*...G.!...0.*
00 21 47 00	B7 ED 42 4E	23 46 60 69	0C CB B9 ED	*.IG...BN#F1....*
49 E9 ED 44	06 4B C3 08	01 ED 6F 06	4F C3 08 01	*I...D.K...o.o...*
F9 06 55 C3	08 01 77 06	5A C3 08 01	F9 06 60 C3	*...U...w.Z...*
08 01 F9 06	66 C3 08 01	00 ED 44 06	6C C3 08 01	*.....f...D.l...*
ED 44 06 73	C3 08 01 34	06 7A C3 08	01 F9 06 82	*.D.s...4.z.....*
C3 08 01 00	06 8A C3 08	01 DD 21 00	00 06 92 C3	*.....!.....*
08 01 DD 21	00 00 06 9B	C3 08 01 ED	4F 06 A5 C3	*...!.....o...*
08 01 34 06	AF C3 08 01	10 FE 1B 7B	B2 C2 9C 00	*.4.....{...*
FD CB 00 7E	20 05 FD 23	C3 66 00 FD	CB 01 7E 20	*.....#f...*
07 FD 23 31	4B 00 18 03	31 49 00 16	0A 01 00 00	*.#1K...1I.....*
ED 49 0B 78	B1 20 FB 15	20 F3 ED 45	03 00 05 00	*.I.x.....E.....*
07 00 08 00	0A 00 0C 00	3A 08 00 05	00 3E 0A 00	*.....:.....*
07 00 AF 35	00 2A 00 02	05 10 05 10	85 03 00 05	*...5*.....*
00 37 3A 0E	20 1C 10 2C	00 A7 47 00	06 00 07 00	*.7:.....G.....*
0F 00 07 00	0E 00 0C 20	0C 20 9C 0A	00 2F 10 1B	*.....:/.....*
10 0C 0A 10	A7 53 17 3A	3F 3C 1F 1C	BA 13 20 15	*.....S.:?<.....*
20 17 00 1A	1C 00 1A 1F	20 1C 20 1A	B7 17 18 3A	*.....:.....*
27 00 27 00	37 95 33 08	00 0A 00 3C	3F 3C 3A B8	*!..7.3.....?<:.*
48 00 0A 00	08 00 06 00	25 00 21 00	B1 45 00 03	*H.....%!.E...*
00 02 00 05	00 0A 00 0C	00 5E 10 BA	0A 00 08 00	*.....^.....E...*
0A 00 08 00	07 20 B3 03	00 03 00 25	00 23 00 28	*.....%#.#.(.*
00 47 20 03	00 03 00 25	00 23 00 2A	00 C8 46 00	*.G.....%#.*.F.*
14 13 14 36	3F 3B A6 15	12 50 12 23	15 0E 20 0E	*...6?;...P.#...*
20 BA 28 00	03 00 23 10	38 00 25 00	05 00 25 40	*...(.#.#.%@.*
27 00 08 00	2A 10 27 10	28 00 0C 00	BB 4A 00 07	*!...*.'.(...J...*
10 0A 10 07	10 0A 10 08	10 A8 0C 00	0B 00 0C 20	*.....*
0C 00 0B 00	0C 20 0C 00	0B 00 0C 20	0A 20 98 03	*.....*
00 03 00 28	00 2C 00 08	00 08 00 2A	00 2D 00 03	*...(.;.....*.-.*
00 07 00 3A	00 0C 00 2A	00 A8 03 00	05 00 27 00	*.....*.....!.*
07 00 0A 00	28 00 08 00	0C 00 1A 00	0C 0A 00 08	*.....(.....*.*
00 A7 21 00	46 00 08 00	0B 00 0A 00	08 00 2D 00	*.!.F.....*.*
AD 28 00 4D	00 08 00 25	00 26 00 48	00 0A 00 A8	*.(M...%.&H...*
23 00 4F 00	0E 00 2F 00	2C 00 4D 00	0C 00 2D 00	*#.0.../...M...-.*
27 00 0A 00	08 6A 00 6A	00 0A 00 3C	30 0A 00 2C	*!...j.j...<0...*
00 0A 00 0C	00 2A 00 09	00 B8 0C 20	2A 20 07 00	*.....*.....*
05 00 03 00	2C 00 2B 00	AC 07 00 05	00 03 00 05	*.....+.....*
00 07 00 05	00 03 00 05	00 27 00 0C	20 27 00 8C	*.....!.....*
03 00 05 00	27 00 27 00	08 00 05 00	1C 0A 00 AA	*.....!.....*
23 00 03 00	07 00 2A 00	2A 00 2A 00	0C 00 0A 00	*#.....*.*.....*
27 00 A3 3F	3E 3C 3A 3C	18 17 25 00	A3 80 44 49	*!..?><.<.%...DI*
45 53 45 53	20 50 52 4F	47 52 41 4D	4D 20 57 55	*ESES PROGRAMM WU*
52 44 45 20	56 4F 4E 20	53 54 45 46	46 45 4E 20	*RDE VON STEFFEN*
4C 45 48 4D	41 4E 4E 2C	20 31 31 34	32 20 42 45	*LEHMANN, 1142 BE*
52 4C 49 4E	2C 20 42 52	4F 44 4F 57	49 4E 45 52	*RLIN, BRODWINER*
20 52 49 4E	47 20 33 30	20 47 45 53	43 48 52 49	*RING 30 GESCHRI*
45 42 45 4E	2E 20 44 49	45 20 57 45	49 54 45 52	*EBEN. DIE WEITER*
47 41 42 45	20 41 4E 20	44 52 49 54	54 45 20 49	*GABE AN DRITTE I*
53 54 20 45	52 4C 41 55	42 54 2C 20	57 45 4E 4E	*ST ERLAUBT, WENN*
20 44 49 45	20 55 52 48	45 42 45 52	53 43 48 41	*DIE URHEBERSCHA*
46 54 20 4E	49 43 43 54	20 56 45 52	41 45 4E 44	*FT NICHT VERBAND*
45 52 54 20	4F 44 45 52	20 56 45 52	53 43 48 4C	*ERT ODER VERSCHL*
45 49 45 52	54 20 57 49	52 44 2E 20	FF FF FF FF	*EIERT WIRD.....*
FF FF FF FF	FF FF FF FF	FF FF FF FF	FF FF FF FF	*.....*
FF FF FF FF	FF FF FF FF	FF FF FF FF	FF FF FF FF	*.....*
FF FF FF FF	FF FF FF FF	FF FF FF FF	FF FF FF FF	*.....*

Bild 3: Programmisting für die Melodieklengel mit Mikroprozessorsteuerung

türlich können auch weniger, dafür aber längere Liedermelodien einprogrammiert werden. Die Speicherung der Töne erfolgt dabei byteweise, wobei ein Byte (8 Bits) einen in Tonhöhe und -länge variablen Ton darstellt. Den genauen Aufbau eines Tonbytes zeigt Bild 2. Die unteren vier Bits repräsentieren die Tonhöhe, wobei die Kombination 0000 eine Ausnahme macht. Diese erzeugt keinen Ton, sondern eine Pause. Folglich ergibt die

Folge 0001 den tiefsten, 1111 den höchsten Ton. Die nächsten drei Bits bestimmen die Tonlänge in Achtelnoten. Hier bedeuten 000 eine Achtelnote, 111 acht Achtelnoten. Längere Töne sind aus mehreren Bytes zusammensetzen. Das Bit 7 wird als Stop-Bit verwendet. Es kennzeichnet, wenn es gesetzt ist, ein Liedende. Folgen zwei Bytes mit gesetztem Stop-Bit aufeinander, ist das Ende der Liedertabelle erreicht.

Die Arbeitsweise der Klingel ist etwa folgende: Bei Betätigen des Klingelknopfes wird über das Speicher-Flip-Flop (D4.2) im Prozessor eine Programmunterbrechung erzwungen, welche diesen aus seinem Warteprogramm herausholt. Ein internes Register (IX) zeigt bereits auf den Anfang des nächsten Liedes. Bevor mit der Erzeugung der Töne begonnen wird, muß zunächst einmal die Anzahl der Tonschwingungen für den nächsten auszugebenden Ton errechnet werden. Dafür enthält das Programm eine Tabelle, die für jeden der 15 möglichen Töne die Schwingungszahl für eine Achtelnote bereithält. Durch eine einfache Multiplikation mit der Anzahl der Achtelnoten steht nun die Zahl der auszugebenden Tonschwingungen fest.

Ein raffiniert ausgeklügeltes Programm übernimmt nun die Ausgabe der Halbschwingungen über das Ausgabe-Flip-Flop (D4.1) in zeitlich richtigem Abstand. Nach Ausgabe aller Halbschwingungen wird der nächste Ton des Liedes ermittelt und ausgegeben, so lange, bis ein gesetztes Stop-Bit erreicht ist. Wird dieses auch noch im folgenden Byte erkannt, erfolgt das Zurücksetzen des Zeigerregisters auf den Anfang der Liedertabelle. Es schließt sich eine Zeitschleife an, die das sofortige Auslösen der Klingel am Liedende bei ungedulden Gästen verhindert. Nun wird das Speicher-Flip-Flop (D4.2) zurückgesetzt, und die Klingel ist zur Ausgabe des nächsten Liedes bereit.

Die Stromversorgung der Schaltung gestaltet sich bei Einsatz eines 2-Kbit-EPROMs (U 2716 C, K 565 RF 2, 2716) ausgesprochen einfach, da nur eine Betriebsspannung von +5 V erforderlich ist. Beim Einsatz kleinerer EPROMs müssen zusätzliche Spannungen bereitgestellt werden, so z. B. +12 V und -5 V beim U 555 D.

Für den Anschluß des Klingelknopfes wurden zwei Varianten entwickelt, damit sowohl einfache Kontakte als auch spannungsliefernde Klingelleitungen (Potentialtrennung vom Netz beachten!) verwendet werden können. Bild 3 zeigt das Programmlisting für die Klingel. Der Abgleich der Schaltung beschränkt sich auf das Einstellen der Prozessor-Taktfrequenz mittels R1 auf 2457,6 kHz. Nach Einschalten der Betriebsspannung und folgendem Power-On-Reset (RC-Kombination am RESET-Eingang des Prozessors) muß der HALT-Ausgang des Prozessors Low-Pegel führen. Andernfalls liegt ein Schaltungsfehler oder ein Bauelemente-Defekt vor. Zur Erleichterung des Nachbaus wurde eine Leiterplatte entworfen (Bild 4), deren Bestückung in Bild 5 gezeigt ist. Diese läßt sich noch um die Stromversorgung erweitern, wodurch man ein kompaktes Gerät erhält.

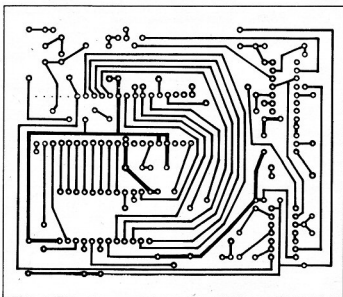
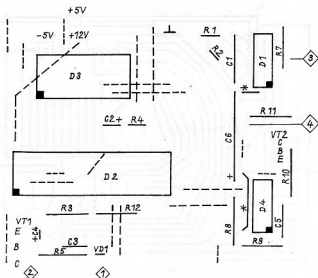


Bild 4: Leitungsführung der Leiterplatte für die Melodieklingsel (Platinengröße 106 mm x 90 mm)



* Stützkondensator 10nF... 33nF
--- Drahtbrücke

Bild 5: Bestückungsplan für die Leiterplatte der Melodieklingsel