

Baby-Telefon

Ohne Elektromog

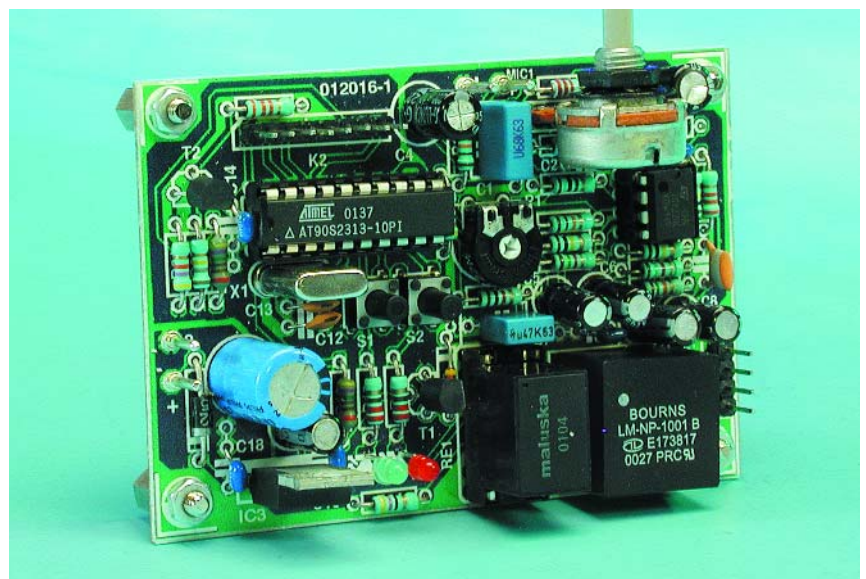
Von Thomas Finke

Herkömmliche Babyfone haben zwei entscheidende Nachteile: Ihre Reichweite ist begrenzt und sie belasten die Umgebung des Spröblings mit Elektromog. Die hier vorgestellte Schaltung beseitigt diese Probleme, in dem sie eine vorher eingestellte Rufnummer anwählt, sobald das Kind schreit.

Babyfone gibt es in vielfältigen Ausführungen. Herkömmliche Schaltungen benutzen entweder als Lichtnetz oder den Äther als Kommunikationskanal, und beides ist mit Nachteilen behaftet. Die Reichweite der Funkmodelle ist beschränkt, die Lichtnetzmodelle dagegen sind darauf angewiesen, dass der Empfang über die gleiche Phase erfolgt. Beide Varianten sind nicht gerade störungsarm und darüber hinaus verseuchen sie ihre Umgebung mit Elektromog. Dies alles vermeidet das hier vorgestellte Baby-Telefon: Es ist an eine Telefonsteckdose angeschlossen und detektiert per Mikro, wenn das Baby schreit. In diesem Fall wählt es eine voreingestellte Telefonnummer (zum Beispiel das Handy der Eltern). Man muss überhaupt nicht den Hörer aufnehmen, denn auf dem Display des angerufenen Apparats erscheint die Telefonnummer des Festanschlusses, gefolgt von einem SOS als Nachwahl. Dann weiß man, dass man im Kinderzimmer erwartet wird ...

Geräuscherkennung

Das Babygeschrei erreicht die Schaltung (**Bild 1**) über die Elektret-Mikrofonkapsel MIC1. Sie wird über R1 mit Betriebsspannung versorgt, wobei R7 und C4 die Mikrospannung zusätzlich entkoppeln. C1 koppelt das NF-Signal aus. Dieses erreicht den Operationsverstärker IC2A am invertierenden Eingang. Beim TS922IN handelt es sich um einen Rail-to-Rail-Doppelopamp von ST, der bei Farnell erhältlich ist. Dies ist auch notwendig, damit der Verstärkungsfaktor $1 - R6/R5$ extrem hoch



eingestellt werden kann. Außerdem wurde aus gleichem Grund die Versorgungsspannung für das IC deutlich erhöht.

Am Schleifer von P2 steht das Audio-Signal zur Verfügung, zwar nicht gerade in High-End-Qualität, aber das ist ja auch nicht erforderlich. C2 verhindert hochfrequente Schwingungen.

Der Spannungsteiler, der aus R2, R3, P1, R4 und dem stabilisierenden C3 besteht, hebt den Gleichspannungsanteil des Signals auf 2,5 V an. So kommt der Opamp mit einer einfachen Versorgungsspannung aus.

Das durch P2 abgeschwächte Signal wird nun zu einem Komparator geführt, der sich nun nicht - wie man im ersten Moment glauben könnte - hinter IC2B verbirgt, sondern im Mikrocontroller AT90S2313. Die beiden Anschlüsse PB0/AIN0 (Pin 12) und PB1/AIN1 (Pin 13) können nämlich als Analogkomparator konfiguriert werden. Wie im Innen-Blockschaltbild 2 des Controllers zu sehen, besitzt der Komparator keinen externen Ausgang, sondern wird nur als interne Interruptquelle verwendet. Dies ist ideal für unsere Anwendung!

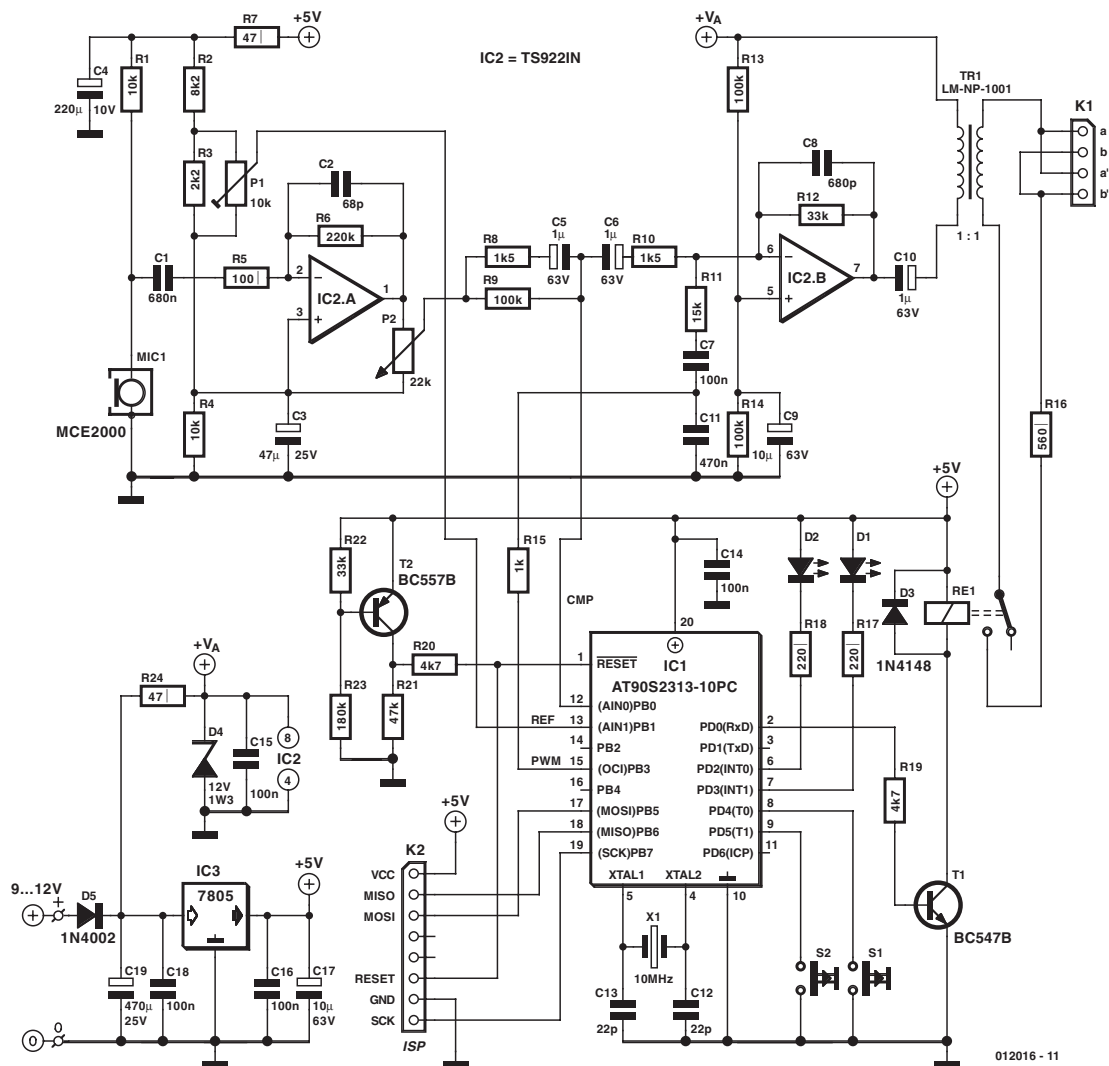


Bild 1. Analoge und digitale Elektronik arbeiten zusammen.

Am Komparatoreingang AIN1 liegt eine Referenzspannung REF, die mit P1 eingestellt werden kann, an AIN0 das NF-Signal CMP. Steigt die NF über die Referenzspannung, wird ein Interrupt im Controller erzeugt. Die Software schaltet AIN0 von Ein- auf Ausgang und legt diesen auf Null. Damit wird das Audiosignal kurzgeschlossen (wobei R8 eine Strombegrenzung darstellt) und statt dessen ein DTMF-Signal (PWM) auf die Signalleitung gelegt. Wie dies funktioniert, wird im Kasten genau erläutert.

Ob Audio- oder PWM-Signal wird einem weiteren Verstärker IC2B zugeführt, der aber nur noch für eine moderate Anhebung sorgt, etwa zweifach für das DTMF- und ungefähr zehnfach für das Audiosignal. R13 und R14/C9 stellen den „Arbeits-

punkt“ auf die halbe Betriebsspannung (von VA) ein, so dass eingangs erwähnte Aussteuerung erreicht werden kann. Die Bandbreite beider Verstärkerstufen beträgt mehr als ausreichende 10 kHz.

Brown-Out-Protection

Die Brown-Out-Protection verhindert, dass der Controller undefinierte Befehle ausführt, wenn die Betriebsspannung zu niedrig ist, also beispielsweise beim Ein- und Ausschalten. Dies ist erforderlich, weil unter anderem die Zielrufnummer im EEPROM steht, wo sie vor unkontrollierten Schreibzugriffen nicht sicher ist. Statt der üblicherweise eingesetzten IC-Lösung wurde hier eine diskrete mit Transistor T2 gewählt, wie sie einer Atmel Appli-

cation-Note entnommen wurde. Sie ist preiswerter und vor allem leichter erhältlich. Die Schaltung aktiviert die Reset-Leitung des Controllers, sobald die Betriebsspannung zu niedrig wird.

Telefonwahl

Über den an K1 angeschlossenen TAE-N-Stecker wird die Schaltung zwischen Telefon und Amtsleitung eingeschleift. Der Controller kann über T1 und Re1 gewissermaßen den Hörer abheben, wobei der Schleifenstrom zwischen a und b über R16 und die Sekundärwicklung des Übertragers TR1 fließt, einem einigermaßen gut erhältlichen Standard-Typ von Bourns. In die Primärseite wird das vom Controller durch Pulsweitenmodulation erzeugte DTMF-Signal eingespeist, die Wahl kann stattfinden. Das Gerät trennt ein an der gleichen TAE-Dose angeschlossenes Telefon nie (wie etwa ein Fax

oder Modem) vom Telefonnetz. So kann man ein einfaches (Reed-) Relais ohne Umschaltkontakt verwenden, außerdem lassen sich am angeschlossenen Telefon die DTMF-Töne kontrollieren.

Dies und das

Die beiden Taster S1 und S2 steuern alle Funktionen, die LEDs D1 und D2 zeigen jeden Status an. Die Bedienung ist zugegeben etwas spartanisch, besonders, wenn es um die Eingabe der Zielrufnummer geht. Da dies jedoch nur sehr selten geschehen muss, scheint die Lösung vertretbar, eine Zehner-tastatur nebst LCD-Display wäre jedenfalls zu viel des Guten.

Die Stromversorgung erfolgt klassisch mit Steckernetzteil (9...12 V), Verpolschutzdiode D5, Lade/Glättungskondensatoren und +5-V-Festspannungsregler. Der Opamp erhält die nicht geregelte Spannung des Steckernetz-teils, die keinesfalls mehr als 12 V liefern darf. D4 ist nämlich nur eine Sicherung gegen die eventuell höhere Leerlaufspannung und sollte keinesfalls dauernd eine höhere Spannung „verheizen“.

Eine Versorgung aus dem Telefonnetz wäre wohl auch möglich gewesen, jedoch scheint eine saubere galvanische Trennung wegen der recht hohen Spannungen angeraten, schließlich handelt es sich um ein Babyfon. Die Taktfrequenz des Controllers liegt mit 10 MHz eigentlich recht hoch, sie wird aber benötigt, um das DTMF-Signal mit der geforderten Frequenzgenauigkeit zu erzeugen. Ein entsprechendes Programmierkabel voraus-gesetzt, kann der Controller über K2 in der Schaltung programmiert werden. Die Pinbelegung entspricht dem von Lattice vorgese-henen Programmierkabel für ihre ispCPLDs. So können mit dem gleichen Kabel ATMEL- und Lattice-Bausteine programmiert werden.

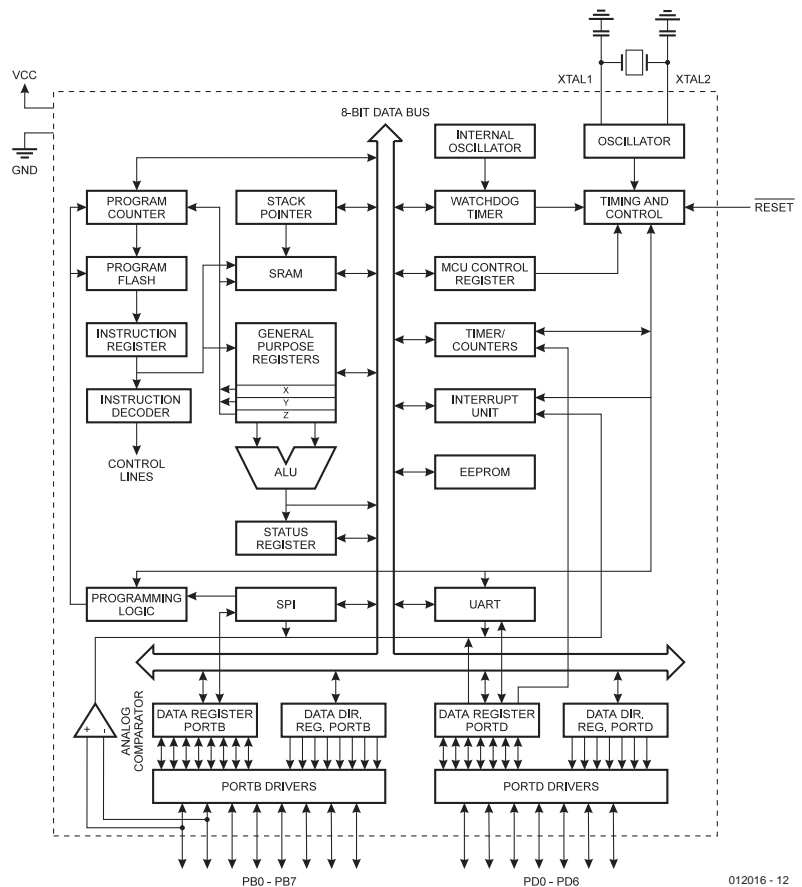


Bild 2. Der Controller verfügt über einen internen Komparator.

Sollen nur ein fertig programmierter Controller Verwendung finden, kann man K2 natürlich weglassen.

Aufbau, Abgleich und Funktionen

Das Layout der einseitig beschichteten Platine in **Bild 3** ist erfrischend kompakt und kann in ein ebenso

kompaktes Kunststoffgehäuse verpackt werden. Zunächst bringt man aber alle Bauteile (von klein nach groß) auf der Platine an. Drahtbrücken gibt es keine, ebenso wenig Besonderheiten bei den Lötarbeiten zu beachten. Für die beiden Impulstaster sollte man Typen mit längerem Tastknopf wählen, so dass man den Controller programmieren kann,

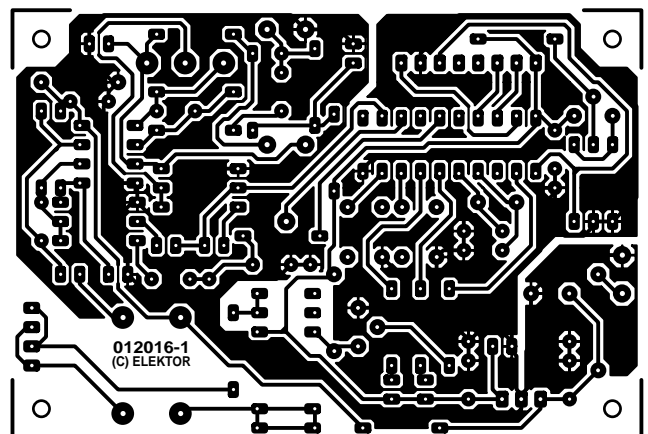
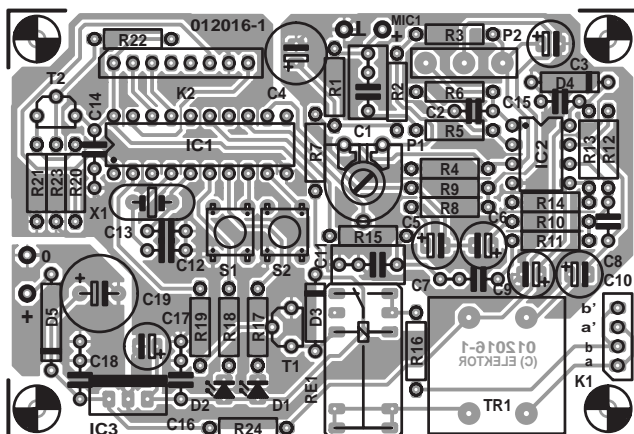


Bild 3. Das Layout der kompakten und einseitigen Platine des Baby-Telefons.

Erzeugung des DTMF-Signals

Einfache DTMF-Generatoren (wie etwa der TP5088 von NS) sind praktisch nicht mehr erhältlich. Aktuelle ICs sind mit einer Reihe weiterer Features ausgestattet, die hier nicht benötigt werden. Sie sind schwer zu beschaffen und man braucht zu ihrer Ansteuerung eine Vielzahl von Signalleitungen (der MT80444 braucht noch das $\Phi 2$ -Signal aus der 6502-Steinzeit). Daher lag der Ansatz nahe, das Signal gleich per Software zu erzeugen.

DTMF ist die Abkürzung für Duale Tone MultiFrequency Signaling. Jeder Wählton besteht dabei aus einem Paar überlagerter Sinus-Töne, einer aus einem Quartett tiefer Töne, der andere aus einer Vierergruppe hoher Töne. Bei der Erzeugung dieser acht Töne müssen alle auf 1,5 % genau getroffen werden, damit die Vermittlung sie eindeutig erkennt.

Leider verfügt der verwendete Atmel-Controller über keinen D/A-Wandler, der vorhandene PWM-Generator erfüllt den Zweck aber genauso gut. Er wird mit 8 bit Auflösung betrieben, woraus sich eine Frequenz von 19,6 kHz ($f_{CK} / 510$) ergibt. Die Breite des PWM-Signals wird vor jeder Periode auf Werte eingestellt, die aus Tabellen abgelesen werden. Das Spektrum eines solchen Signals weist neben den Spitzen bei den Frequenzen der beiden Sinus Töne einen Peak bei $f = 0$ auf (das Signal ist natürlich nicht gleichspannungsfrei, aber dafür sind ja C7/C11 da). Weitere Frequenzanteile liegen erst wieder bei der PWM-Frequenz, da diese aber vom einfachen Tiefpass aus R15 und C11 ausgefiltert werden, bekommt die Vermittlung davon nichts mit und hört tatsächlich nur die beiden Wählfrequenzen. Das geforderte Amplitudenverhältnis zwischen hoher und niedriger Tongruppe wurde direkt in den Sinustabellen berücksichtigt. War die Frequenzgenauigkeit mit einer gespeicherten Periode nicht zu erreichen, wurden zwei oder drei abgelegt, wodurch insgesamt etwa 200 Bytes mit Sinus-Tabellen belegt sind.

Stückliste

Widerstände:

R1,R4 = 10 k
 R2 = 8k2
 R3 = 2k2
 R5 = 100 Ω
 R6 = 220 k
 R7,R24 = 47 Ω
 R8,R10 = 1k5
 R9,R13,R14 = 100 k
 R11 = 15 k
 R12,R22 = 33 k
 R15 = 1 k
 R16 = 560 Ω
 R17,R18 = 220 Ω
 R19,R20 = 4k7
 R21 = 47 k
 R23 = 180 k
 P1 = 10 k Trimpoti liegend
 P2 = 22 k linear, kleine Monoversion

Kondensatoren:

C1 = 680 n
 C2 = 68 p
 C3 = 47 μ /25 V stehend
 C4 = 220 μ /10 V stehend
 C5,C6,C10 = 1 μ /63 V stehend
 C7,C14...C16,C18 = 100 n keramisch
 C8 = 470 p
 C9,C17 = 10 μ /63 V stehend
 C11 = 470 n
 C12,C13 = 22 p
 C19 = 470 μ /25 V stehend

Halbleiter:

D1 = LED, rot, 3 mm
 D2 = LED, grün, 3 mm
 D3 = 1N4148
 D4 = Z-Diode 12 V/1W3
 D5 = 1N4002
 T1 = BC547B
 T2 = BC557B
 IC1 = AT90S2313-10PC
 (programmiert EPS 012016-41)
 IC2 = TS922IN (bei Farnell)
 IC3 = 7805

Außerdem:

K1 = 1-4-poliger Pfostenverbinder
 K2 = 1-8-poliger Pfostenverbinder
 S1,S2 = Impulstaster 6-6 mm²
 Re1 = Subminiaturrelais 16-119-11,5 mm³ mit SPDT-Kontakt (z.B. Maluska FRS1B-S DC 5V, (5 V, 56 Ω , Conrad 505188)
 X1 = Quarz 10 MHz (mit $C_{load} = 32$ p parallel)
 Tr1 = Line-Übertrager (Bourns LM-NP-1001 B)
 MIC1 = Elektret-Mikrofonkapsel (z.B. Monacor MCE2000)
 Platine EPS 012016-1
 Diskette EPS 012016-11 (C-Quelle- und Hexkode)
 (siehe Service-Seiten in der Heftmitte)

ohne sich dabei die Finger zu verrenken.

Die Mikrokapsel bringt man aus Sicherheitsgründen im Gehäuse unter. Damit sie auch das Babygeschrei auffangen kann, sollte man das Gehäuse an dieser Stelle perforieren. Das TAE-Kabel wird mit einem Knickschutz fest mit dem Gehäuse verbunden.

Sind alle Lötarbeiten kontrolliert, kann das Gerät eingeschaltet werden. Die grüne LED leuchtet als Einschaltkontrolle. Mit Taster S1 kann man das Gerät deaktivieren, die grüne LED verlischt. Wird im aktiven Zustand ein Geräusch erkannt, so blinkt die grüne LED. Zunächst beachten wir die LED aber nicht, sondern stellen die Verstärkung an P2 so ein, dass ein Audiosignal in der maximalen Lautstärke (spricht: lautes Babygeschrei) IC2B nicht übersteuert. Dies lässt sich mit einem Oszilloskop gut überprüfen. Anschließend kann mit P1 die passende Empfindlichkeit eingestellt werden. Nicht jedes Geräusch löst gleich einen Alarm aus, vielmehr muß es eine bestimmte Frequenz überschreiten und eine vorgegebene Mindestzeit andauern (Zeit und Frequenz sind fest im EEPROM abgelegt und bei Bedarf zu ändern).

Sind diese Bedingungen erfüllt, zieht Re1 an; der "Hörer" wird abgenommen. Die rote LED zeigt Dauerlicht. Anschließend wird die eingestellte Rufnummer angewählt und nach einigen Sekunden Wartezeit (zwei bis drei Klingelzeichen) wieder aufgelegt. Anschließend bleibt das Gerät für einige Zeit unscharf, (rote LED blinkt) um ständige Anrufe zu vermeiden. Schließlich wechselt das Baby-Telefon wieder in den Ausgangszustand.

Röchelfunktion

Ein Druck auf S2 wählt (bei aktivem Gerät) die programmierte Rufnummer sofort an. Parallel zu S2 lässt sich ein Relais-Kontakt oder ein offener Kollektor eines anderen Gerätes (etwa Alarmanlage, Funkempfänger) schalten, das den Anruf auslösen soll.

Einstellen der Rufnummer

Das Programmieren der Rufnummer ist nicht eben komfortabel, da sie aber nach dem Ausschalten dauerhaft im EEPROM gespeichert bleibt, ist es nur selten erforderlich. Durch langes (>2 s) Drücken von S1 bei aktivem Gerät gelangt man in den Programmiermodus (rote LED ein, grüne aus). Jetzt muß die erste Ziffer der Rufnummer eingestellt werden, indem man entsprechend oft S1 betätigt (bei 0 wird gar nicht gedrückt). S2 bestätigt die Eingabe und wechselt zur nächsten Stelle. Auf diese Weise wird die komplette Nummer "eingetastet" und die Programmierung durch abermaliges langes Drücken von S1 abgeschlossen. Stellt man eine oder mehrere Zif-

fern der Rufnummer auf 10, wird hier bei der Wahl eine Pause eingelegt, was beim Anschluss an Nebenstellenanlagen nötig sein kann (Amtsholung). Passiert bei der Eingabe ein Fehler, kann man durch zweimaligen langen Druck auf S1 von vorne beginnen und die vorhergehende Eingabe überschreiben.

Abfragen der Rufnummer

Bei so viel Tipperei tut eine abschließende Kontrolle Not. Deshalb kann man durch langen Druck auf S2 die Programmierung abfragen. Wieder zeigt die rote LED Dauerlicht. Die grüne blinkt so oft, wie es der ersten Ziffer entspricht und wiederholt dies nach kurzer Pause (bei 0 bleibt sie natürlich dunkel). Mit S2 schaltet man zur nächsten Ziffer, nach der letzten verlischt die rote LED, die grüne leuchtet auf: Das Baby-Telefon ist betriebsbereit.

Programmierung

Das komplette Programm zur Steuerung des Baby-Telefons ist in C geschrieben. Damit Konstanten (Mindestfrequenz der Geräusch-

erkennung, diverse Zeiten) auch ohne Compiler geändert werden

gerät auch direkt ins EEPROM schreiben:

| Adresse | Funktion | Default |
|------------------------------------|---|---------|
| \$00-\$1F | Zielrufnummer im ASCII-Code, \$00 ist das Endzeichen | |
| \$20 | Grenzfrequenz, oberhalb der Geräusch als Babygeschrei klassifiziert wird ($f_g = X \cdot 38 \text{ Hz}$) | 10 |
| \$21 | Länge des Überwachungszeitfensters in $2^X \cdot 6,5 \text{ ms}$ | 10 |
| \$22 | So viele „laute“ Zeitabschnitte á 6,5 ms müssen im Überwachungszeitfenster liegen, dass Alarm ausgelöst wird | 80 |
| \$23 | Zeitintervall zwischen Wahl und Auflegen (Klingelzeichen) $X \cdot (\text{ca.}) 6 \text{ s}$ | 4 |
| \$24 | Gerät bleibt nach jeder Wahl $X \cdot 16 \text{ s}$ unscharf, bevor bei andauerndem Geräusch neu gewählt wird | 10 |
| X ist der Inhalt der Speicherzelle | | |

können, stehen diese im EEPROM des Mikrocontrollers. Auch die Zielrufnummer wird natürlich im EEPROM untergebracht, man kann sie mit einem Programmierkabel/

Sind beim Einschalten S1 und S2 gedrückt, werden die Defaultwerte ins EEPROM geschrieben.

(012016)rg

Anzeige