



Best.-Nr.: 92448
Version 1.00
Stand: Januar 2011

wakeboX – Universal-Anzeige- und -Bedienterminal USB-BAT

Technischer Kundendienst

Für Fragen und Auskünfte stehen Ihnen unsere qualifizierten technischen Mitarbeiter gerne zur Verfügung.

ELV • Technischer Kundendienst • Postfach 1000 • D-26787 Leer

Reparaturservice

Für Geräte, die aus ELV-Bausätzen hergestellt wurden, bieten wir unseren Kunden einen Reparaturservice an. Selbstverständlich wird Ihr Gerät so kostengünstig wie möglich instand gesetzt. Im Sinne einer schnellen Abwicklung führen wir die Reparatur sofort durch, wenn die Reparaturkosten den halben Komplettbausatzpreis nicht überschreiten. Sollte der Defekt größer sein, erhalten Sie zunächst einen unverbindlichen Kostenvoranschlag.

Bitte senden Sie Ihr Gerät an: **ELV • Reparaturservice • Postfach 1000 • D-26787 Leer**

ELV Elektronik AG • Postfach 1000 • D-26787 Leer
Telefon 0491/6008-88 • Telefax 0491/6008-244



Wohlfühlen mit wakeboX – Universal-Anzeige- und -Bedienterminal USB-BAT

Die wakeboX ist eine Softwarelösung, die das FS20- und FHT-System mit umfangreichen Audio- und Automationsfunktionen vereint. Zuverlässig, flexibel, komfortabel und ohne Stressfaktoren geweckt zu werden – das und vieles mehr macht die wakeboX möglich. Die Server-Client-basierte Software steht unter der GNU General Public Licence (GPL) für jedermann zur freien Verfügung.

Ein universelles Bedien- und Anzeigeterminal wird vollständig von der wakeboX-Software unterstützt und entspricht in seiner Funktionalität dem originalen wakeboX-Terminal. Es ist darüber hinaus aber auch für andere Anwendungen einsetzbar, da es über eine HID-USB-Schnittstelle verfügt.

wakeboX – den Morgen genießen

Wollen Sie mit einem Sonnenaufgang, dem Duft von frischem Kaffee und Ihrer Lieblingsmusik geweckt werden? Das ist, grob zusammengefasst, der Hintergrund der Entwicklung der wakeboX-Software, zu der

sich vor einiger Zeit Frank Wittmann, Andreas Lohr, Dieter Walz und Erik Rull aus ganz praktischen Gründen veranlasst sahen.

Es folgt der (redaktionell bearbeitete) Bericht der Entwickler von wakeboX, die im Hause ELV großen Anklang fand und in der Folge zu diesem Artikel führte. Im ersten Teil werden die Geschichte, die Installation und die Möglichkeiten des wakeboX-Systems vorgestellt. Das von ELV entwickelte universelle Bedien- und Anzeigeterminal wird den Schwerpunkt des zweiten Artikels bilden.

Wie es begann

Im Winter 2005 entstand die Idee. Frank Wittmann, technikbegeisterter Hobby-Entwickler, wollte der alltäglichen „Aufweck-Qual“ ein Ende setzen und den Tag entspannt beginnen: Eine Tageslicht-Leuchtstoffröhre, ein dazu passender FS20-Funk-Dimmer, eine FS20-USB-Hauszentrale (FHZ 1000 PC) und der Win-

Display:	Vollgrafikdisplay (122 x 32 Pixel)
Bedienelemente:	2 Taster, 1 Inkrementalgeber mit integriertem Taster
Software-Schnittstelle:	Generic USB-HID, offengelegtes Protokoll
Hardware-Schnittstelle:	USB-Mini-B-Buchse
Spannungsversorgung:	USB-powered
Stromaufnahme:	max. 90 mA
– Displaybeleuchtung an:	ca. 70 mA
– Displaybeleuchtung aus:	ca. 20 mA
– Standby:	ca. 3,3 mA
Abmessungen (B x H x T):	182 x 60 x 76 mm

dows Media Player samt Wecker-Plug-in eines Drittanbieters dienten dabei als Lösungsansatz Nummer 1. Dabei wurden schon nach kurzer Zeit zwei grundlegende Probleme offensichtlich:

1. Frank musste immer zwei Timer programmieren und diese dann morgens auch entsprechend wieder abschalten. Den einen für das Licht, den anderen für den Sound – nicht besonders bequem.

2. Windows stürzte beängstigend oft ab, so dass Frank – als Notnagel und im Parallelbetrieb – immer noch auf einen handelsüblichen Wecker zurückgreifen musste.

So kam es, dass Frank Wittmann sich mit Programmierer Andreas Lohr und Tontechniker Dieter Wald zusammenschloss und zunächst eine Interessengemeinschaft mit dem Namen „Mr. Wake“ gründete, später wurde daraus „wakeboX“ [1]). In Eigenregie konzipieren, programmieren, konstruieren, designen und bauen – so lauteten die Statuten der Gemeinschaft. Zunächst gab es den Einwand, dass es doch schon unzählige Licht-MP3-Wecker gäbe, von namhaften Herstellern fertig gebaut und akribisch getestet. Was aus drei allerdings nicht zufrieden gestellt hat, war das bei diesen Geräten stets vorhandene Prinzip „Kompromisslösung“: Plastik, soweit das Auge reichte, Klangqualität nicht vorhanden, nicht erweiterbar und beschränkt auf nur eine Lichtquelle als Aktor.

2009 gab es die erste veröffentlichte Version von wakeboX. Zu dieser Zeit ist Erik Rull als weiterer Programmierer zum Team hinzugestoßen und hat seither mit vielen Erweiterungen die wakeboX bereichert. Was ist die wakeboX nun?

Funktionsübersicht

Hauptmerkmale der wakeboX sind die Haussteuerung, die Audioausgabe, die Ablaufsteuerung, die Eingabeverarbeitung und die Ereignisplanung (Timer).

Haussteuerung

Die Haussteuerung wird mit Hilfe des FS20- und des FHT-Systems realisiert. Es können beliebig viele Empfänger ins System integriert werden, welche individuell mit allen jeweils verfügbaren Parametern konfigurierbar bleiben.

Darüber hinaus werden zusätzliche Features wie beispielsweise „Dimmen über Zeitspanne“ oder „Eimalschalter“ (darauf kommen wir noch) bereitgestellt.

Audioausgabe

Die Audioausgabe ist ein wesentliches Merkmal der wakeboX, da die gezielte akustische Stimulation beim Wecken einen großen Einfluss auf das Aufwachgefühl hat.

Hierbei können MP3-Dateien und MP3-Internetstreams – auch über Wiedergabelisten (.m3u oder .pls) – wiedergegeben werden. Es sind mehrere MP3-Player zeitgleich nutzbar, sie können auf eine gemeinsame Soundkarte ausgegeben. Darüber hinaus unterstützt die wakeboX mehrere Soundkarten, so dass für verschiedene Räumlichkeiten eine individuelle Akustik realisierbar ist.

Alle Parameter, wie zum Beispiel die Lautstärke, sind für jeden Player individuell einstell- und auch automatisiert änderbar.

Ablaufsteuerung

Die Ablaufsteuerung ist ein Makroprozessor, der in der Lage ist, mehrere Aktionen (Haussteuerung, Audiowiedergabe etc.) in geordneter Reihenfolge abzuwickeln. Dabei wird jede Aktion zu einem bestimmten Zeitpunkt ausgelöst, der sich aus dem Startzeitpunkt des Ablaufs plus einer vorgegebenen Zeitspanne errechnet.

Die Eingabeverarbeitung ist für das Entgegennehmen und Zuordnen von Eingabeereignissen (FS20-Sender, Multimedia-IR-Fernbedienungen etc.) zuständig. Dabei kann jede Eingabe (Tastendruck) beliebigen Aktionen (Haussteuerung, Audiowiedergabe etc.) oder gar ganzen Abläufen zugeordnet werden.

Ereignisplanung

Die Ereignisplanung ist der „Wecker“ der wakeboX. Sie ist für das zeitgesteuerte Auslösen von Abläufen zuständig. Ein Ereignis wird dabei durch eine konkrete Uhrzeit, eine Menge an Wochentagen und den auszulösenden Ablauf definiert. Darüber hinaus können abgespeicherte Ereignisse individuell aktiviert und deaktiviert werden.

In der Praxis lassen sich so beliebig viele Ereignisse definieren, jedoch nur die gerade benötigten werden eingeschaltet und dann auch ausgeführt. In Kombination mit einem installierten NTP-Client (NTP: Network Time Protocol) erhält man zudem über das Internet eine stets genaue Systemzeit, wodurch ein Nachstellen der aktuellen Uhrzeit entfällt.

Systemübersicht und Hardwareanforderungen

Die wakeboX-Software besteht aus einem in C++ geschriebenen Server und einem zugehörigen, in Java geschriebenen grafischen Client. Die Kommunikation zwischen Client und Server erfolgt dabei über das Netzwerkprotokoll TCP/IP.

Der wakeboX-Server ist für das freie Betriebssystem GNU/Linux entwickelt worden. Die Hardware spielt dabei eine untergeordnete Rolle, da GNU/Linux auf den verschiedensten Hardwarearchitekturen wie beispielsweise x86, ARM etc. lauffähig ist. Geeignet sind daher insbesondere kleine, stromsparende PCs wie der „Smart-Mini-PC“ von ELV (Bild 1) oder noch kleinere Embedded PCs wie z. B. der SheevaPlug [2], der in einem Steckernetzteilgehäuse Platz findet und auf Monitor und Tastatur komplett verzichtet. Bild 2 zeigt den nur ca. 110 x 70 x 49 mm kleinen Winzling. Auch die neue ELV-Linux-Entwicklungs-Plattform LCU 1 ist für diese Aufgabe geeignet.

Der wakeboX-Client ist für die Konfiguration und Kontrolle des Servers zuständig und ist durch die Laufzeitumgebung von Java im Gegensatz zum Server auch unter anderen Betriebssystemen lauffähig. Der Client kann somit auf jedem

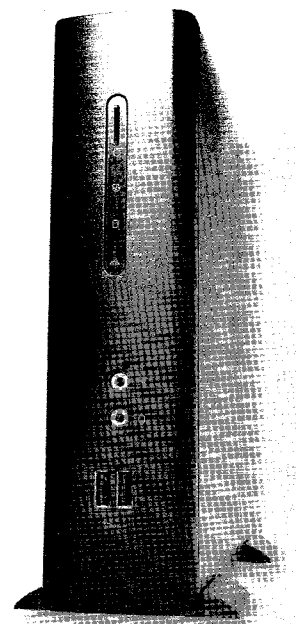


Bild 1: Mini-PCs wie der Smart-Mini-PC sind eine ausreichende Hardware-Grundlage für das wakeboX-System (ELV-Best.-Nr. JD-782-92).

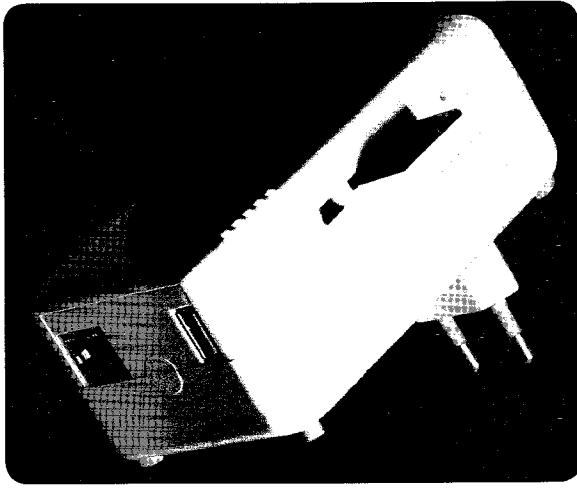


Bild 2: Absolute Mini-Formos – der USB-Plug

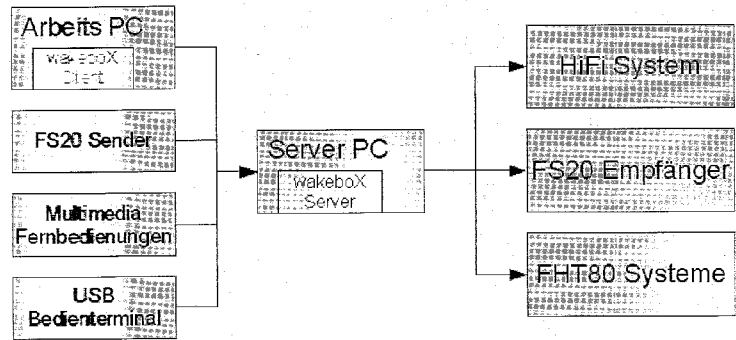


Bild 3: Die Systemübersicht über die Hardware des wakeboX-Systems

gängigen Arbeits-PC oder Mac gestartet werden.

Zum Ansteuern der FS20- und FHT-Komponenten wird eine FHZ 1000 PC oder FHZ 1300 PC benötigt, dazu für die Audioausgabe eine Soundkarte. Letztere kann entweder die Onboard-Soundkarte des Server-PCs oder eine externe Soundkarte für eine bessere Klangqualität sein. Die externe USB-Soundkarte „Sound Blaster Play“ von Creative [3] beispielsweise bietet eine schöne Lösung für wenig Geld. Für anspruchsvollere Ohren hingegen eignet sich die etwas teurere USB-Soundkarte „Transit“ von M-Audio [4].

Ab Version 0.72.3 werden auch der Sender FS20 PCS und der Empfänger FS20 PCE unterstützt und bilden eine kostengünstige Alternative zur FHZ, wenn nur das FS20-System zum Einsatz kommt.

Weitere optionale Komponenten sind Multimedia-Infrarot-Fernbedienungen sowie das USB-HID-Bedienteil, das im nächsten „ELVjournal“ detailliert vorgestellt wird. Bild 3 zeigt die Hardware-Übersicht über das gesamte wakeboX-System.

Installation

Voraussetzung für eine Installation des Servers ist ein installiertes GNU/Linux-System. Hier empfiehlt sich die Verwendung einer Standarddistribution wie „Ubuntu“ [5], „Debian“ [6], „SuSE“ [7] oder „Fedora“ [8].

Eine Anleitung zur Installation dieser Systeme ist auf den jeweiligen Seiten der Anbieter zu finden. Weiterhin müssen einige Pakete zum Kompilieren des Servers auf dem GNU/Linux installiert werden, was über die benutzerfreundlichen Paketverwaltungsprogramme der Distributionen recht einfach ist.

Benötigte Pakete sind der Compiler „GNU C++ (g++)“, das Steuerungssystem „GNU Make“, die Entwicklungsumgebung für ALSA (Advanced Linux Sound Architecture) sowie der Audioplayer „mpg123“. Einzelheiten zu den einzelnen Paketen sind auf den Hilfeseiten von [1] zu finden. Der wakeboX-Server ist im Archiv „wakebox- x.xx.tgz“ enthalten und kann über [9] heruntergeladen werden. Es sollte die neueste verfügbare Version verwendet werden. Das Archiv kann anschließend in einer Shell entpackt werden mit dem Befehl:

```
$ tar xvzf wakebox-x.xx.tgz
```

Dabei steht x.xx für die entsprechende Versionsnummer.

Zum Kompilieren des Servers wechselt man nun in das soeben entpackte Verzeichnis und dort in das Unterverzeichnis „server“:

```
$ cd wakebox-x.xx
$ cd server
```

Dort ist das durch „autoconf“ erzeugte Konfigurationskript „configure“ aufzurufen, welches die Voraussetzungen zum Kompilieren des Servers prüft und ein entsprechendes Makefile erstellt:

```
$ ./configure
```

Sollte eine Wetterstation (z. B. WS 300) mit wakeboX betrieben werden, so ist stattdessen aufzurufen:

```
$ ./configure --enable-libftdi
```

Hierdurch wird die Verfügbarkeit der Bibliothek „libftdi“ überprüft und die entsprechenden Quellen zur Unterstützung der Wetterstation zum Kompilieren vermerkt.

Nachdem das Konfigurationskript fehlerfrei durchgelaufen ist, kann nun der Server kompiliert werden mittels:

```
$ make
```

Je nach Prozessor und Taktgeschwindigkeit kann dies etwas Zeit in Anspruch nehmen.

Bevor man wakeboX installiert, ist sicherzustellen, dass „mpg123“ entweder als Paket der Distribution oder durch Herunterladen, Kompilieren und Installieren von [10] verfügbar ist.

Nun erfolgt das Installieren des kompilierten Servers auf dem System. Dies muss als Administrator (root) geschehen. Dazu ist der Benutzer zu wechseln. Für die meisten GNU/Linux-Distributionen mit:

```
$ su
```

Bzw. für Distributionen wie „Ubuntu“, in denen ein separates root-Passwort nicht vorgesehen ist, mit:

```
$ sudo su
```

Jetzt ist der Benutzer als „root“ angemeldet und die Installation kann durchgeführt werden via:

```
# make install
```

Das Installationskript stellt hierbei einige Fragen:

```
Path to mpg123? [ /usr/local/bin/
mpg123 ]
```

Dies ist der Pfad zur ausführbaren Datei „mpg123“. Ist ein Vorschlag in eckigen Klammern verfügbar, kann man diesen durch Drücken der Eingabetaste übernehmen. Sollten die eckigen Klammern leer sein, so ist das ein Hinweis, dass „mpg123“ noch nicht auf dem System installiert oder die ausführbare Datei zumindest nicht in den Standard-Suchpfaden zu finden ist. In diesem Fall ist die Installation über „Strg-C“ abzubrechen und sicherzustellen, dass „mpg123“ verfügbar ist. Ohne „mpg123“ wird die Wiedergabe von MP3-Dateien und Internetradio nicht möglich sein.

```
Automatically start wbxid with system startup? [y/N]
```

Dies ist die Frage, ob der Server (auch bekannt als „Systemdienst“) beim Hochfahren des Betriebssystems automatisch gestartet werden sollte. Es ist empfehlenswert, hier mit „Ja“ (Taste „y“) zu antworten. Ansonsten muss der wakeboX-Server nach jedem Neustart manuell gestartet werden, um dessen Funktionen nutzen zu können.

```
The configuration files are stored in /usr/local/etc/wbxid and /usr/local/etc/init.d/wbxid. Do you want to install links in /etc to easily access these files? [y/N]
```

Dies ist die Frage, ob Verknüpfungen im Verzeichnis „/etc“ angelegt werden sollen, damit man leicht zu den etwas versteckteren Installationspfaden „/usr/local/etc/wbxid“ und „/usr/local/etc/init.d/wbxid“ gelangen kann. Wird diese Frage mit „Ja“ („y“) beantwortet, was prinzipiell zu empfehlen ist, kann man über „/etc/init.d/wbxid“ den Systemdienst starten, stoppen oder neu starten bzw. über „/etc/wbxid“ das Konfigurationsverzeichnis erreichen. Der Server kann nun durch

```
# /etc/init.d/wbxid start
```

bzw.

```
# /usr/local/etc/init.d/wbxid start
```

gestartet werden.

Der Client ist in der Programmiersprache Java geschrieben und benötigt daher die „Java-Laufzeitumgebung JRE – Java Runtime Environment“. Für MS Windows, aber auch für GNU/Linux und andere Betriebssysteme kann die Java-Laufzeitumgebung direkt vom Hersteller [11] heruntergeladen und relativ einfach unter Befolgen der Anweisungen installiert werden.

Eine weitere Möglichkeit unter GNU/Linux ist die Installation über das Paketverwaltungssystem der Distribution. Weitere Informationen hierzu finden sich auf den Hilfeseiten von [1]. Nun ist der vorkompilierte Client „WakeboxClient-x.xx.jar“ über [9] herunterzuladen, wobei auch hier die neueste Version verwendet werden sollte. Der Client wird unter Windows anschließend einfach per Doppelklick gestartet. Unter GNU/Linux erfolgt der Start des Clients über das Kommando:

```
java -jar WakeboxClient-x.xx.jar
```

Nach Start des Clients kann dem Client nun durch Rechtsklick auf die Weckkugel (oben links) die IP-Adresse des Servers mitgeteilt werden. Wenn der Server auf einem anderen Rechner (beispielsweise Embedded PC) läuft, ist hier die IP-Adresse dieses anderen Rechners einzutragen. Wenn der Server hingegen auf dem gleichen System läuft wie der Client, lautet die IP-Adresse „127.0.0.1“. Dies ist die „Loopback“-Adresse, die auf denselben Rechner abbildet.

Der Standard-Port lautet in beiden Fällen 5250. Mit einem Linksklick auf die Weckkugel wird die Verbindung zum Server aufgebaut. Der Kontakt ist hergestellt, sobald die blaue „LED“ in der Weckkugel leuchtet und die Uhrzeit in der oberen Anzeige des Clients angezeigt wird.

Anpassung und Konfiguration

Um die wakeboX an die eigene Hardwarekonfiguration anzupassen, ist mit dem Werkzeugsymbol („weitere Einstellungen“) in das Einstellungs Menü zu gehen und anschließend das Untermenü „Server“ zu wählen (Bild 4). Hier sind alle konfigurierten Komponenten hierarchisch in einer Baumstruktur angeordnet und es ist möglich, neue Komponenten wie FS20-Aktoren (Schalter, Dimmer) hinzuzufügen und alle Komponenten zu parametrieren.

Die Konfiguration sowohl von FS20-Eingabegeräten (Fernbedienungen, Sensoren usw.) als auch von Multimedia-Fernbedienungen hingegen erfolgt im Untermenü „Eingabegeräte“ (Bild 5). Hier kann man Fernbedienungen einzeln definieren und deren Tasten auf Aktionen wie Schaltvorgänge, Musik oder ganze Abläufe abbilden.

Abläufe wiederum können über das Zahnräder-symbol („Abläufe“) im Untermenü „Konfiguration“ (Bild 6) definiert werden. Ein Ablauf ist eine zeitlich definierte Abfolge von Aktionen, wobei alle Zeitangaben innerhalb eines Ablaufs immer relativ sind und sich auf den Start des Ablaufs beziehen. Hier findet die eigentlich kreative Arbeit am wakeboX-System statt, da hier insbesondere auch die Weckabläufe erstellt werden.

Die Ereignisplanung erreicht man über das Wecker-

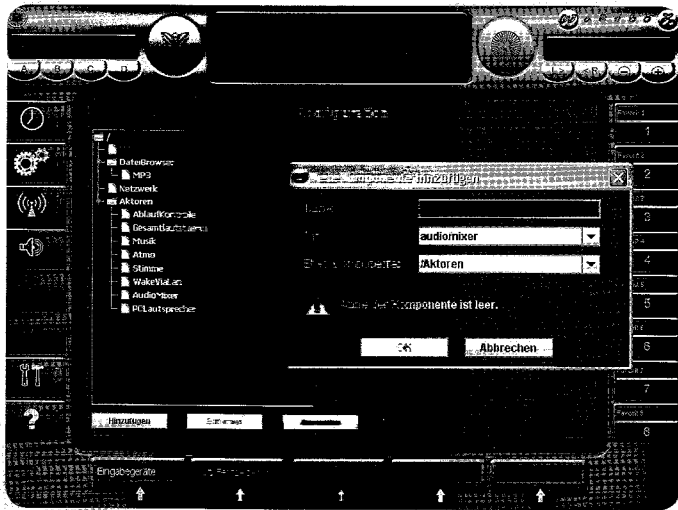


Bild 4: Über das Untermenü „Server“ gelangt man zur Konfiguration der einzelnen Komponenten. Hier lassen sich Komponenten hinzufügen, löschen und parametrieren.

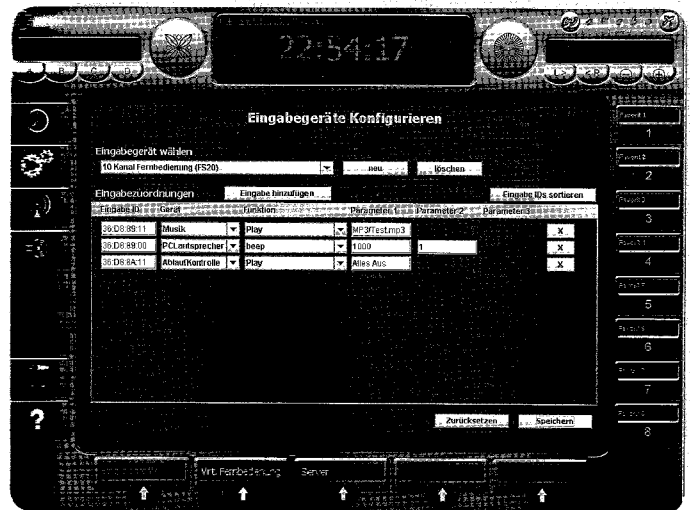


Bild 5: Das Konfigurationsmenü für Eingabegeräte

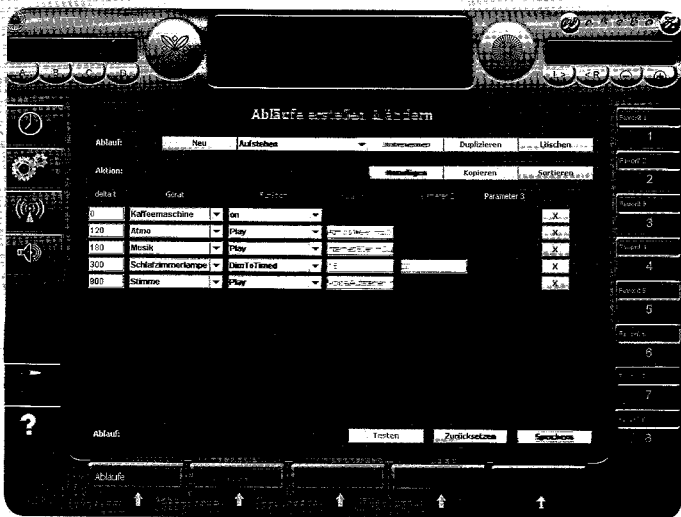


Bild 6: Hier findet die eigentlich kreative Arbeit am wakeboxX-System statt – das Ablauf-Planungsmenü.

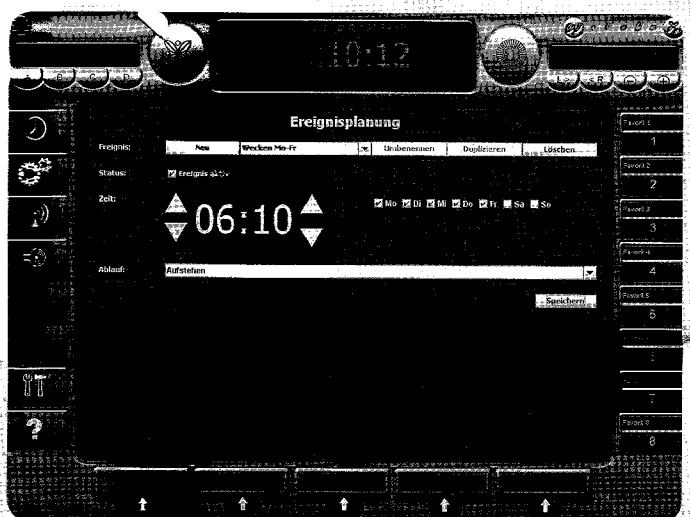


Bild 7: In der Ereignisplanung werden Weckereignisse definiert und verwaltet.

symbol (Bild 7). Hier sind die Weckereignisse zu definieren und werden hier auch verwaltet. Die Definition geschieht durch Angabe einer Uhrzeit, der gewünschten Wochentage und des Ablaufs, der beim Erreichen dieser Zeit gestartet werden soll.

Darüber hinaus können die meisten Komponenten auch unmittelbar gesteuert werden: Zur direkten Steuerung von FS20-Empfängern klickt man auf das Funkensymbol („FS20-Aktoren“, Bild 8). Über das Lautsprecher-Symbol („Audio“) gelangt man zur Steuerung der Audioplayer (Bild 9), und die direkte Ablaufkontrolle ist über das Zahnradsymbol („Abläufe“, Bild 10) im Untermenü „Abläufe“ zu erreichen. Weiterführende Informationen zur Bedienung des Clients sind auf den umfangreichen Hilfeseiten des Projekts zu finden.

Anwendungsbeispiele

Neben der Hauptaufgabe der wakeboxX, dem angenehmen, sanften Wecken, gibt es beliebig viele Anwendungsszenarien. Hier nur ein paar Anregungen:

Der FS20-Einmalschalter an der Kaffeemaschine

Das Starten einer vorbereiteten Kaffeemaschine ist natürlich ein Kernpunkt eines „anständigen“ Weckablaufs. Damit die Kaffeemaschine aber auch nur dann anspringt, wenn sie vorbereitet wurde (Kaffee, Filter,

Wasser), bietet die wakeboxX eine Spezialisierung des FS20-Schalters: den FS20-Einmalschalter. Dieser führt das Anschaltkommando (z. B. durch Ablauf getriggert) nur dann durch, wenn er zuvor scharf geschaltet wurde.

Nach erfolgreichem Einschalten ist dieser Aktor dann erneut scharf zu schalten, bevor man ihn wieder einschalten kann, da die Kaffeemaschine ja dann auch erst wieder vorbereitet werden muss.

Zum Scharf-Schalten eignet sich beispielsweise ein FS20-Aufputz-Wandsender neben der Kaffeemaschine, bei dem ein Taster das Scharf-Schalten („rearm“) des FS20-Einmalschalters betätigt.

Sonnenauf- und -untergang

Ein weiteres Highlight für Aufweckabläufe ist natürlich der simulierte Sonnenaufgang. Dieser wird vom Dimmer-Aktor der wakeboxX durch die Funktion „DimToTimed“ direkt unterstützt. Der Funktion werden hierzu neben der Zieldimmstufe auch die Dimmgeschwindigkeit in Sekunden pro Dimmschritt übergeben. Die Funktion ist natürlich auch gleichermaßen

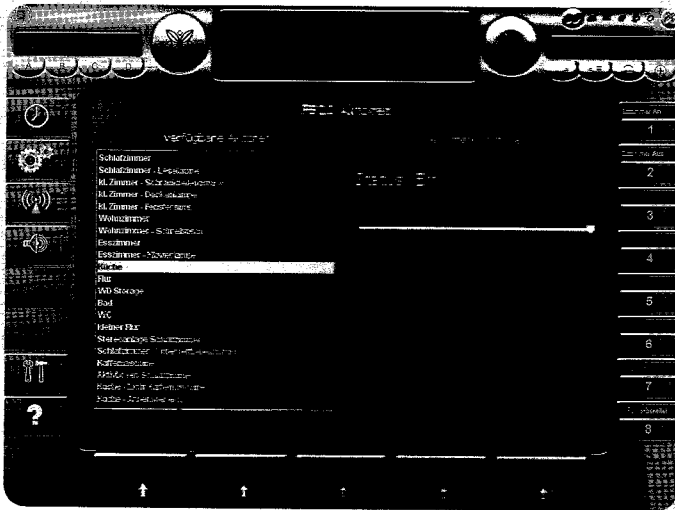


Bild 8: Über dieses Menü sind FS20-Aktionen steuerbar.



Bild 9: Hier kann direkt auf die Audio-Wiedergabesteuerung zugegriffen werden.

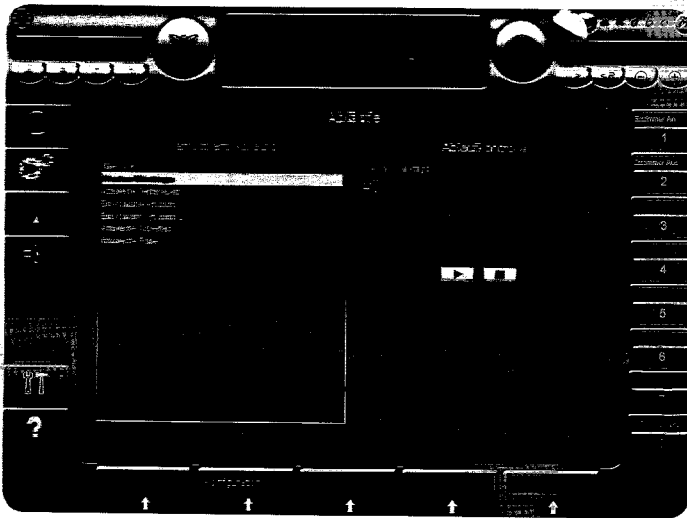


Bild 10: Neben der automatisierten Ablaufsteuerung kann man auch manuelle Abläufe starten, stoppen und deren Status kontrollieren.

umgekehrt als Sonnenuntergang für Einschlafprogramme nutzbar. Eine ähnliche Funktion gibt es außerdem für die Audioplayer (Funktion „FadeVolume-Timed“). Hiermit kann die Song-Lautstärke langsam erhöht oder auch umgekehrt bis zur völligen Stille gesenkt werden.

Alles aus

Eine andere – auch komfortable, ökologische und ökonomische – Anwendung ist das Ausschalten aller Komponenten durch einen Tastendruck. Somit kann beispielsweise neben der Wohnungstür ein FS20-Aufputz-Wandsender montiert werden, den man beim Verlassen der Wohnung einfach drückt, um einen Ablauf auszulösen, der alle konfigurierten Komponenten (z. B. Licht) abschaltet. Ein weiterer FS20-Aufputz-Wandsender neben dem Bett, der denselben Ablauf auslöst, macht es sehr angenehm, alles einfach auszuschalten, wenn man schon im Bett liegt.

Nachtschaltung

Eine Nachtschaltung ist für das nächtliche „Mal-raus-Müssen“ eine tolle Sache. Dazu werden die Lichter im

Schlafzimmer und im Badezimmer mit FS20-Dimmern ausgestattet. Diese werden beim Betätigen einer FS20-Sendetaste dann auf einen für nächtliche Verhältnisse angenehmen Wert gedimmt, so dass man seinen Weg leicht findet, ohne von grellem Licht geblendet zu werden. Eine weitere FS20-Sendetaste schaltet dann wieder alle Lichter aus. Um am nächsten Tag wieder die volle Dimmstufe beim Betätigen des „normalen“ Wandsenders im Bade- und Schlafzimmer zu erhalten (denn die Dimmer merken sich ja die letzte Dimmstufe), kann die wakeboX durch eine Eingabebeurteilung beim Drücken dieses Schalters ein Kommando „volle Dimmstufe“ hinterherschicken.

Fazit und Ausblick

wakeboX wird seit Jahren auf vier Systemen des Entwicklerteams betrieben. Dabei erfreuen sie sich immer noch am Komfort und der Zuverlässigkeit dieses Systems und werden nicht müde, wakeboX kontinuierlich weiterzuentwickeln. Ein weiterer wesentlicher Teil des wakeboX-Systems ist das auf [1] vorgestellte, selbst konstruierte USB-Bedienteil. Dieses kleine, für den Nachttisch bestimmte Terminal mit LCD-Display, Druckknöpfen und Drehrad ist ein nützlicher Helfer, der einen schnellen Zugriff auf alle wesentlichen Funktionen von wakeboX bietet.



Werden Sie geweckt von Ihrer Lieblingsmusik

Tabelle 1: Befehlsübersicht

Befehls-ID	Beschreibung / Bedeutung der Befehls-ID	Nachfolgende Parameter-Bytes
0xF0	Firmware-Version ausgeben	keine weiteren Parameter-Bytes
0xF1	Display-Hinterleuchtung für bestimmte Zeilen einschalten	1 Byte für Zeitdauer in 10-ms-Schritten – Wertebereich von 0 (dauerhaft eingeschaltet) bis 255
0xF2	Display-Hinterleuchtung ausschalten / Beleuchtungsmodus ändern	1 Byte für Beleuchtungsmodus - 1: Displayhinterleuchtung wird bei Nutzereingabe (Taster, Inkrementalgeber) automatisch für 5 Sekunden eingeschaltet - 0: Displayhinterleuchtung wird bei Nutzereingabe (Taster, Inkrementalgeber) nicht automatisch eingeschaltet
0xF3	Piepton mit variabler Zeitdauer	1 Byte für Zeitdauer in 10-ms-Schritten – Wertebereich von 0 (dauerhaft eingeschaltet) bis 255
0xF4	Piepton ausschalten	keine weiteren Parameter-Bytes
0xF5	Taster und Inkrementalgeber abfragen	keine weiteren Parameter-Bytes
0xF6	Inkrementalgeber zurücksetzen	keine weiteren Parameter-Bytes
0xF7	Benutzereingaben automatisch senden	1 Byte zum Ein- oder Ausschalten - 2: eingeschaltet (bei Statusänderung wird gesendet) - 1: eingeschaltet (interne Tastenerkennung) - 0: ausgeschaltet
0xF8	Reset	keine weiteren Parameter-Bytes
0xD0 bis 0xD7	Page 0 bis 7 beschreiben	Displayzeile jeweils pro Byte eine Spalte der Page.3E (keine ungenutzten Bytes)
0xD8	String anzeigen	2 Byte für Zeile und Position: - Byte 1: High-Nibble: Zeile (0-3) - Byte 1: Low-Nibble: Zeichen- (0) oder Pixelposition (1) - Byte 2: Position (0-19 oder 0-121) - 20 Bytes definieren den String - 1 Byte für Stringabschluss (0x00)
0xD9	Displayinhalt löschen	keine weiteren Parameter-Bytes
0xDA	Page löschen	1 Byte zur Seitenauswahl – Wertebereich von 0 (Seite 1) bis 7 (Seite 8)
0xDB	String/Zeile löschen	1 Byte für Seitenauswahl (0-3)
0xDC	Display initialisieren	keine weiteren Parameter-Bytes

Schneller Zugriff auf wakeboX

Im Laufe der Entwicklung des WakeboX-Systems entstand ein kleines Bedienterminal, das etwa vom Nachttisch, vom Schreibtisch oder anderen Orten aus einen schnellen Zugriff auf alle wesentlichen Funktionen von wakeboX bietet. Es ist mit einem universell programmierbaren Grafikdisplay, zwei Tasten und einem Drehgeber ausgestattet und in einem modern gestalteten, kompakten Gehäuse untergebracht. Das im folgenden vorgestellte Universal-Anzeige- und -Bedienterminal USB-BAT entspricht in seiner Funktionalität dem Original.

Doch das USB-BAT ist nicht nur auf den Einsatz im wakeboX-System beschränkt, es kann auch in anderen Anwendungen als Anzeige- und Bedienterminal dienen. Durch die USB-Schnittstelle und die verwendete USB-HID-Klasse kann das Terminal unter den verschiedensten Betriebssystemen und Anwendungen eingesetzt werden. In Tabelle 1 sind die möglichen Befehle zur Ansteuerung des Terminals in Kurzform zusammengefasst.

Displaymodul

Im USB-BAT kommt ein sehr kompaktes und universelles Vollgrafik-Displaymodul mit 122 x 32 Pixel zum Einsatz. Das Modul benötigt eine Versorgungsspannung von 5 V und eine negative Versorgungsspannung für die Kontrasteinstellung. Die Hinterleuchtung kann über die 5-V-Versorgungsspannung des Moduls betrieben werden. Der interne Aufbau ist in Bild 11 zu sehen. Für die Segmentansteuerung sind die beiden Displaycontroller PT6520 zuständig, sie übernehmen jeweils eine Hälfte der Displaysegmente.

Die Firmware des Mikrocontrollers teilt das Display in 8 Seiten mit jeweils 62 Byte ein (siehe Bild 12). Jede Seite kann über den zugehörigen Befehl 0xD0 bis 0xD7 (siehe Befehlsübersicht) direkt beschrieben werden. Die Zuordnung der 62 Parameter-Bytes zu den Displaysegmenten ist Bild 13 zu entnehmen.

Befehlsübersicht

Das USB-BAT hat 11 Befehle (siehe Tabelle 1), mit denen die Bedienelemente (Taster und Inkrementalgeber) abgefragt, der Displayinhalt geändert und der Summer ein- und ausgeschaltet werden. Der Summer kann mit Befehl 0xF3 dauerhaft oder für eine bestimmte Zeitdauer eingeschaltet, mit Befehl 0xF4 wieder ausgeschaltet werden. Gleiches gilt für die Displayhinterleuchtung, dafür stehen die Befehle 0xF1 und 0xF2 zur Verfügung. Um einfachen Text anzuzeigen, ist Befehl 0xD8 notwendig, es können in 4 Zeilen jeweils maximal 20 Zeichen dargestellt werden. Die Position in der Zeile lässt sich dabei buchstaben- oder pixelgenau angeben. Um das Display ganz individuell zu gestalten, können die Befehle 0xD0 bis 0xD7 genutzt werden. Damit kann das gewünschte Pixelmuster seitenweise (siehe Bild 12) ins Display geschrieben werden.

Wenn der gesamte Displayinhalt gelöscht werden soll, dann reicht es, den Befehl 0xD9 an das Display zu senden.

Die Bedienelemente können mit dem Befehl 0xF5 vom PC aus abgefragt werden (Polling). Da kontinuierliches Senden von Befehlen an den USB-BAT die Frequenzerzeugung für den Summer stört, kann das

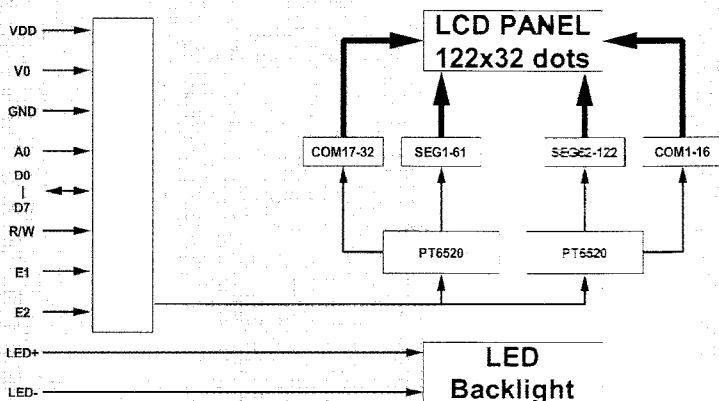


Bild 11: Der Blockschnittplan der Display-internen Ansteuerung

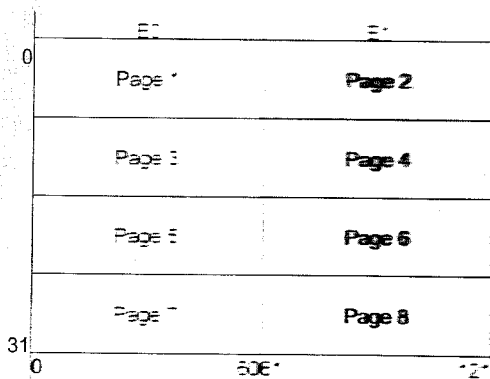


Bild 12: Die Aufteilung des Displays in 8 Seiten

	Byte 0	Byte 60
Bit 0		
Bit 1		
Bit 2		
Bit 3		
Bit 4		
Bit 5		
Bit 6		
Bit 7		

Bild 13: Die Zuordnung der 60 Parameter-Bytes zu den Displaysegmenten

Terminal mit dem Befehl 0xF7 so konfiguriert werden, dass es Statusänderungen der Bedienelemente automatisch an den PC gesendet werden. Es lassen sich zwei Automatik-Modi auswählen. In Modus 1 wird die interne implementierte Tastenerkennung verwendet, um kurzen und langen (ca. 5 Sekunden) Tastendruck zu erkennen. Sobald ein gültiger Tastendruck erkannt bzw. der Inkrementalgeber betätigt wurde, wird die Antwort auf den Befehl 0xF5 an den PC gesendet. Falls die Tastenerkennung am PC erfolgen soll, ist Modus 2 interessant. Ist er aktiv, wird sobald sich der Zustand der Taster oder des Inkrementalgebers ändert eine Meldung an den PC gesendet. Die Abfrage der Taster erfolgt dabei in einem 10-ms-Raster.

Der interne Positionszähler für den Inkrementalgeber des USB-BAT lässt sich mit dem Befehl 0xF6 zurücksetzen. Fehlen noch die Befehle 0xF0, 0xF8 und 0xDC, mit denen man die Firmwareversion abgefragt, einen Neustart veranlassen und eine Initialisierung des Displays starten kann.

Weitere Details und eine ausführliche Befehls- und Protokollbeschreibung sind im Download-Bereich zu finden.

Schaltungsbeschreibung

Hauptkomponente des USB-BAT (Schaltung in Bild 14) ist der Mikrocontroller C8051F326 von Silicon Laboratories, ein aktuelles 8051-Derivat mit integriertem USB-Controller. Der Controller ist USB-2.0-kompatibel und unterstützt den Low- und Highspeed-USB-Modus.

Am Port P 0 sind die Taster TA 1, TA 2 und der Inkrementalgeber DR 1 angeschlossen. Port-Pin P 0.7 steuert das Gate des MOSFET T 2 an, über den die Spannung für die Flipflops, das Display und den Summer geschaltet wird. Mit den Port-Pins P 0.5 und P 0.6 sind die Bausteine IC 2 oder IC 3 auswählbar. Diese sind vom Typ 74HCT174 und enthalten jeweils sechs D-Flipflops. Diese Flipflops sind für die Displaysteuerung notwendig, da mit Port 2 nicht mehr genügend Steuer- und Datenleitungen zur Verfügung stehen.

Als Anzeige kommt ein Vollgrafikmodul mit 122 x 32 Pixel samt Hinterleuchtung zum Einsatz.

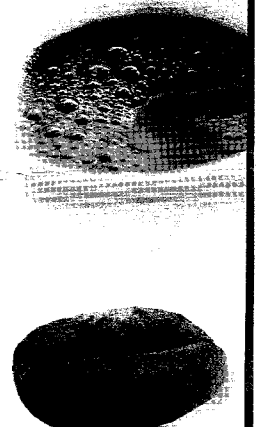
Die Hinterleuchtung lässt sich über den Transistor T 3 ein- und ausschalten. Für die Kontrasteinstellung ist eine negative Spannung zwischen -3,3 V und -1,5 V notwendig, die mit IC 4 erzeugt wird. Über das Potentiometer R 10 ist der Kontrast an die jeweiligen Anforderungen anpassbar. Aufgrund der großen Anzahl der Displaysegmente sind im Displaymodul zwei Displaycontroller verbaut, die man über die beiden Pins E 1 und E 2 auswählen kann. Da trotz des Einsatzes der Flipflops immer noch eine Steuerleitung für den Reset-Pin des Displays fehlt, kommt noch ein NAND-Glied zum Einsatz. Solange keiner oder nur einer der beiden Displaycontroller ausgewählt wurde (E 1 und/oder E 2 liegen auf Low-Pegel), liegt am Reset-Pin des Displays ein High-Pegel an und das Display befindet sich im normalen Betriebsmodus.

Die Spannungsversorgung erfolgt vollständig über die USB-Schnittstelle. Der Mikrocontroller enthält einen Spannungswandler, mit dem die notwendige Versorgungsspannung von 3,3 V intern erzeugt wird. An Pin 6 steht diese Spannung auch für andere Schaltungsteile zur Verfügung. Über den MOSFET T 2 kann der Mikrocontroller die Spannung für die Flipflops, das Display und den Summer abschalten.

Nachbau

Nach der Schaltungsbeschreibung kommen wir nun zum Nachbau des USB-BAT. Der Aufbau besteht aus zwei Platinen, der Basisplatine und der Tasterplatine. Alle Platinenteile sind im Lieferzustand miteinander verbunden und verfügen über einfach abknickbare Sollbruchstellen zum Trennen der Platinen. Vor der Bestückung sind die zwei Platinenteile zu trennen. Durch die Vorbestückung der SMD-Bauteile beschränkt sich der Nachbau auf die korrekte Platzierung und das Verlöten der bedrahteten Bauelemente bzw. Platinenteile. Dennoch ist die Bestückung wie üblich auf Bestückungsfehler oder Lötzinnbrücken zu prüfen. Die Bestückung der restlichen Bauelemente erfolgt in gewohnter Weise anhand des Bestückungsplans, der Stückliste und unter Zuhilfenahme der Platinenfotos.

Als Erstes sind auf der Basisplatine der Widerstand

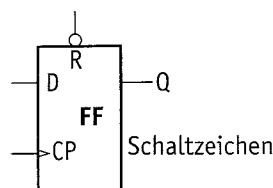
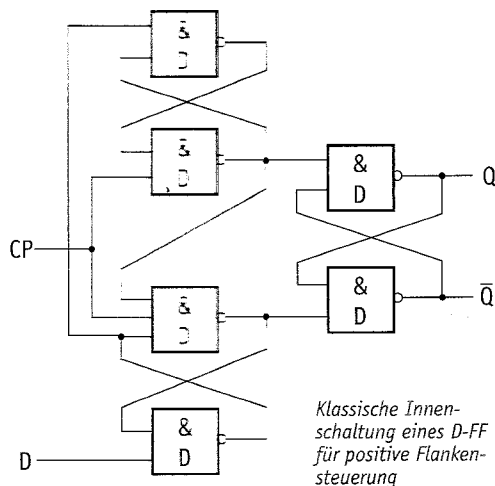


Frischer Kaffee am Morgen – vollautomatisiert dank wakebox!

R 8 und der Elektrolyt-Kondensator C 7 (Polarität beachten, liegende Position) zu bestücken und zu verlöten. Es folgen das Potentiometer R 10 und danach der Sound-Transducer PZ 1 (Polarität beachten). Nun werden die vier zweipoligen Stiftleisten ST 1 bis ST 4 sowie die beiden neunpoligen Stiftleisten bestückt und eingelötet. Die neunpoligen Stiftleisten bilden die Aufnahme für das Display. Im nächsten Schritt ist zuerst die USB-Buchse von der Lötseite aus zu bestücken und zu verlöten, erst danach kann das Display bestückt werden, da man sonst die USB-Buchse nicht mehr verlöten kann. Der letzte Schritt auf der Basis-

platine ist die Bestückung des Inkrementalgebers DR 1. Auf der Tasterplatine sind die beiden Taster TA 1 und TA 2 zu bestücken, danach kann die gesamte Tasterplatine auf die bereits verbauten Stiftleisten ST 1 bis ST 4 gesteckt und von oben verlötet werden. Dabei ist darauf zu achten, dass die Orientierung der Tasterbeschriftung mit der Beschriftung des Potentiometers R 10 übereinstimmt (Bild 15).

Nun erfolgt der Einbau in das Aluminiumgehäuse. Dazu sind zuerst die vier Senkkopfschrauben M2 x 14 mm durch die Schraubenlöcher auf der Rückseite des Gehäuses einzusetzen. Danach ist die Rückseite des



Wahrheitstabelle eines flankengesteuerten D-Flipflop

/R	CP	D	Q_n	Beschreibung
1	0	X	Q_{n-1}	Speichern
1	⎓	0	0	Rücksetzen
1	⎓	1	1	Setzen
0	X	X	0	Reset

Wahrheitstabelle eines zustandsgesteuerten D-Flipflop

/R	CP	D	Q_n	Beschreibung
1	0	X	Q_{n-1}	Speichern
1	1	0	0	Rücksetzen
1	1	1	1	Setzen
0	X	X	0	Reset

Der D-Flipflop und seine Anwendung

Das D-Flipflop (D-FF) ist ein modifiziertes RS-Flipflop, bei dem der Set- und Reseteingang zusammengelegt (D) wurde. Damit nicht beide Eingänge das selbe Signal erhalten, wird einer der Eingänge negiert ausgeführt. Zudem hat das D-FF noch einen Takteingang (CP), es arbeitet daher nur taktgesteuert (synchron). Darauf bezieht sich auch das „D“ im Namen. Es steht für Delay und soll angeben, dass Änderungen am Dateneingang D nicht am Ausgang übernommen werden, solange keine Änderung am Takteingang erfolgt.

Das D-FF kann weiter in taktzustands- oder taktflankengesteuert unterschieden werden. Sie unterscheiden sich darin, zu welchem Zeitpunkt bzw. bei welchem Zustand des Taktsignals der Eingangszustand am Ausgang übernommen wird. Bei dem taktzustandsgesteuerten D-FF wird er übernommen, wenn am Takteingang eine 1(High) anliegt, das taktflankengesteuerte D-FF übernimmt den Eingang nur bei einer positiven Taktflanke. Eine inverse Ausführung bei 0 (Low) am Takteingang oder einer negativen Flanke wäre möglich.

In den beiden Wahrheitstabellen sind alle möglichen Signalzustände erfasst und erläutert.

Das taktzustandsgesteuerte D-FF speichert den bisherigen Ausgangszustand (Q_{n-1}) und gibt ihn solange am Ausgang (Q_n) aus, bis am Takteingang eine 1 anliegt. Dann wird je nach Zustand des Eingangs der Ausgang gesetzt oder zurückgesetzt. Während

der Takteingang 1 ist, werden alle Änderungen des Eingangs sofort am Ausgang übernommen.

Auch beim taktflankengesteuerten D-FF ändert sich der Ausgang nicht, solange der Takteingang auf 0 (Low) liegt. Im Gegensatz zum zustandsgesteuerten D-FF wird der Eingangszustand aber nur bei einer positiven Flanke am Ausgang übernommen. Während der folgenden 1(High)-Phase werden keine weiteren Zustandsänderungen des Eingangs übernommen.

Aufgrund der Speichereigenschaft können D-FFs in Schaltungen eingesetzt werden, um vorhandene IO-Leitungen mehrfach zu nutzen (Multiplexing). Ein 74HC(T)174 stellt 6 D-FFs bereit, die alle über eine gemeinsame Taktleitung angesprochen werden können. Schließt man nun 2 ICs parallel an einen 8-Bit-breiten Mikrocontroller-Port an, können 6 Leitungen doppelt genutzt werden. Eine der beiden restlichen Leitungen ist jeweils an den Takteingang der ICs anzuschließen. Nun können die Daten nacheinander an den Portpins angelegt und ein Taktimpuls für das gewünschte IC erzeugt werden, um die Daten zu übernehmen.

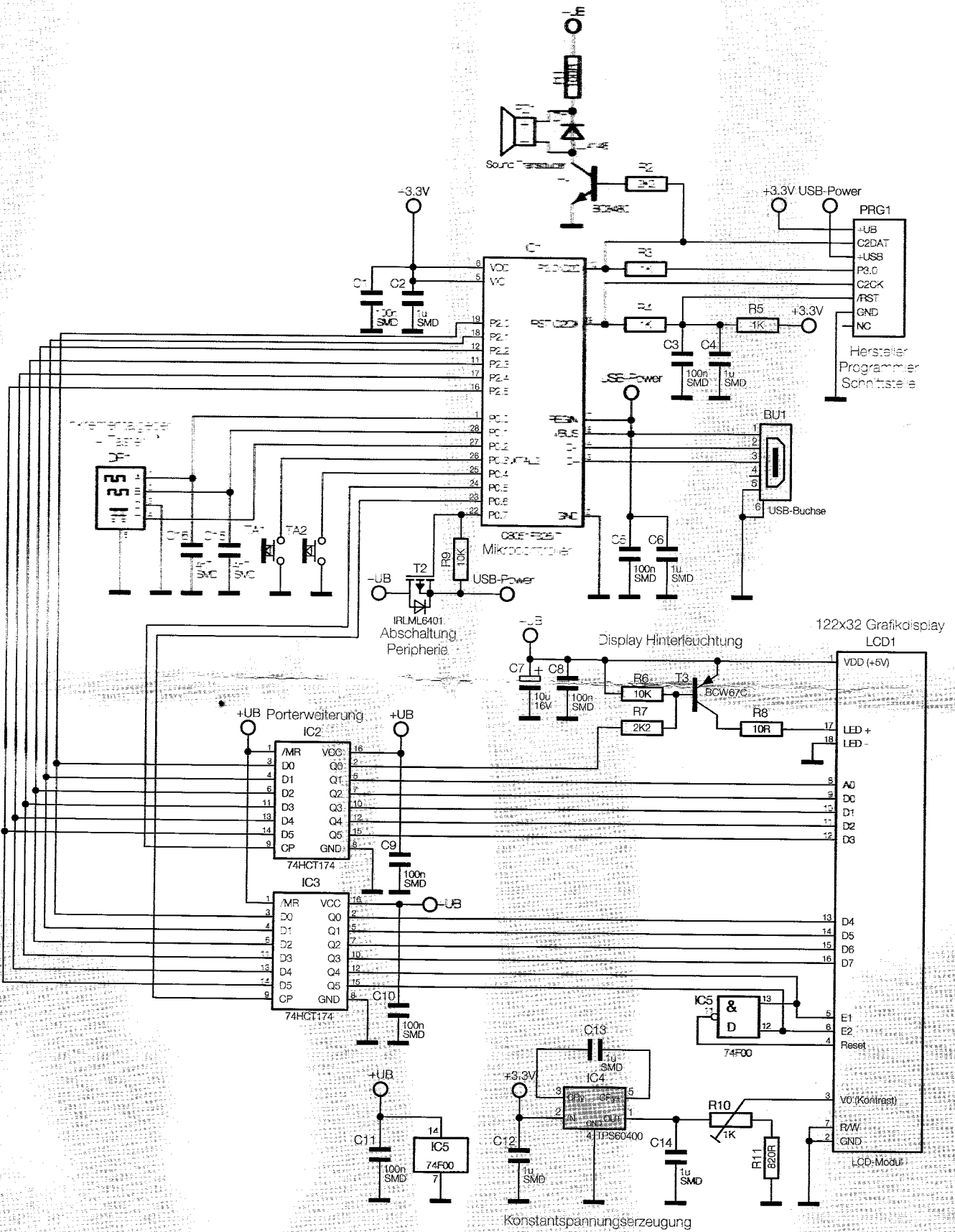


Bild 14: Die Schaltung des wakeboX-Terminals USB-BAT

Gehäuses mit den vier eingesetzten Schrauben auf eine ebene Fläche zu stellen. So kann man nun von oben in das Gehäuse auf die vier Schrauben schauen. Jetzt sind die vier 8-mm-Distanzrollen auf die Schrauben zu setzen und im Anschluss die Platine auf die Distanzrollen aufzusetzen. Mit den beiliegenden Scheiben und Muttern wird die Platine nur leicht befestigt, da zunächst ein prüfender Blick auf die Rückseite des Gehäuses klären muss, ob die USB-Buchse passgenau in der vorgesehenen Öffnung liegt. Ist dies der Fall, können die Schrauben fest angezogen werden. Zur Kontrolle kann Bild 16 dienen. In der Querschnittaufnahme ist die Position der verschiedenen Bauteile zu erkennen.

Nun ist die Frontblende an der Reihe. Da die transparente Displayscheibe schon eingeklebt ist, kann die Frontblende sofort eingesetzt werden. Am einfachsten

geht dies, wenn zuerst der obere Rand der Frontblende auf die kleine Metallleiste des Gehäuses gelegt und dann der untere Rand mit leichtem Druck eingesetzt wird (Bild 17). Es folgt das Aufsetzen des Drehknopfes. Jetzt sind die beiden Standfüße mit den vier EJOT-Schrauben (2,5 x 12) an das Gehäuse zu montieren (Bild 18). Davor sind noch jeweils 2 Gehäusfüße unter die Standfüße zu kleben, um ein Wegrutschen des USB-BAT beim Bedienen zu verhindern.

Die Schrauben dürfen nicht zu fest angezogen werden, damit sich der Neigungswinkel des USB-BAT noch einstellen lässt. Wenn der gewünschte Neigungswinkel gefunden ist, können die Schrauben später noch fester angezogen werden, so dass sich das Display nicht mehr von allein verstellen kann. Damit ist der Nachbau komplett fertiggestellt und die Inbetriebnahme kann erfolgen.

Inbetriebnahme

Nachdem die Gehäusefüße montiert und eingestellt wurden, verbindet man das Terminal über ein USB-Kabel mit einem PC. Das Display sollte

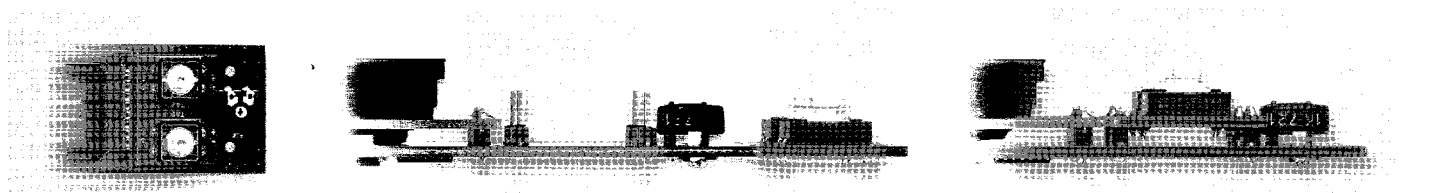
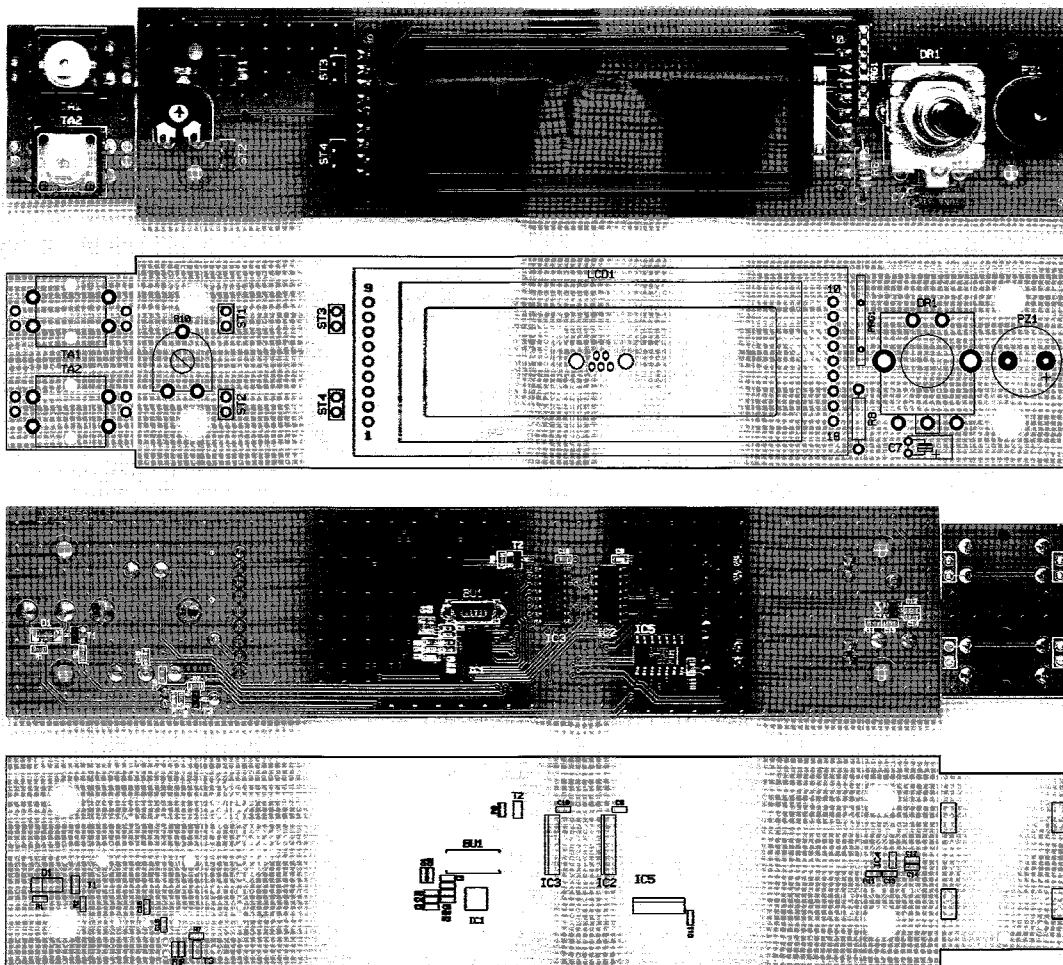


Bild 15: Die Platine als Gesamtstück (links) sowie vor (Mitte) und nach (rechts) Abbrechen und Auflöten der Tasterplatine



Ansicht der bestückten Platinen des USB-BAT mit zugehörigem Bestückungsplan, hier ist die Tastenplatine noch nicht abgetrennt. Oben die Oberseite mit montiertem Display, unten die Unterseite

nun „Bedien-Anzeige-Terminal USB-BAT VX.X“ anzeigen. Dank der HID-Schnittstelle wird das Terminal automatisch erkannt und installiert, es ist kein zusätzlicher Treiber notwendig.

Im Gerätemanager ist das Terminal unter dem Punkt „Eingabegeräte (Human Interface Devices)“ zu finden. Mit der USB-BAT-Testsoftware aus dem Download-Bereich können die Grundfunktionen sofort getestet werden.

Für den Einsatz als Bedienterminal im wakebox-System muss mindestens die Version 0.7.1 des wakebox-Servers installiert sein. Erst ab dieser Version kann das ELV USB-BAT als Alternative zum original wakebox-Bedienterminal eingesetzt werden.

Falls das Terminal in anderen Anwendungen eingesetzt werden soll, kann die vollständige Schnittstellenbeschreibung heruntergeladen werden.

Anregungen und Beispiele finden sich im Quellcode der Testsoftware im Downloadbereich. **ELV**

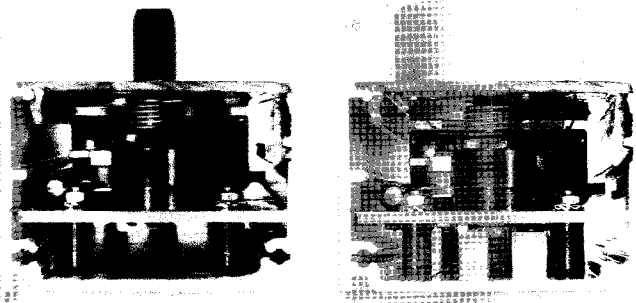


Bild 16: Zusammenbau des Terminals, rechts mit aufgesetztem Drehknopf für den Drehgeber

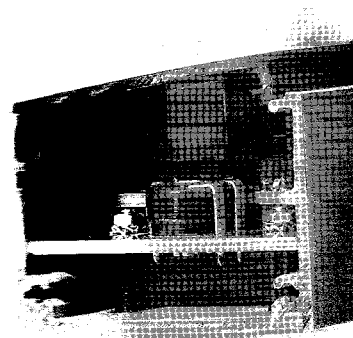


Bild 17: Einsetzen der Frontplatte

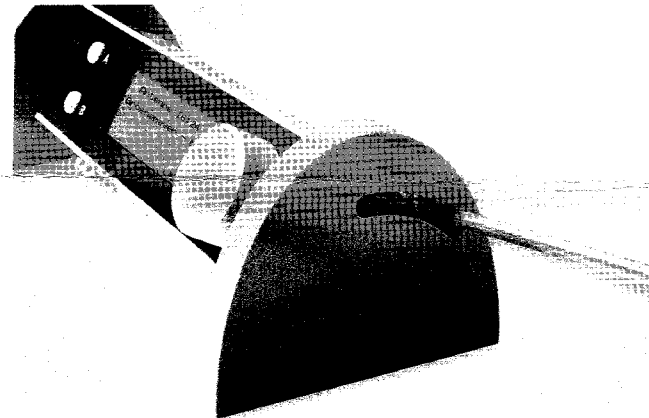


Bild 18: Standfüße anschrauben

Weitere Infos:

- [1] www.wakebox.de
- [2] <http://de.wikipedia.org/wiki/SheevaPlug>
- [3] <http://de.store.creative.com/products/product.aspx?catid=1&pid=17892>
- [4] http://de.m-audio.com/products/de_de/Transit.html
- [5] www.ubuntu.com
- [6] www.debian.com
- [7] <http://de.opensuse.org>
- [8] <http://de.fedoraproject.org/de>
- [9] www.wakebox.de/download.html
- [10] www.mpg123.de
- [11] www.java.com/de/download/manual.jsp

Widerstände:

10 Ω/1 W	R8
100 Ω/1 %/SMD/0603	R1
820 Ω/1 %/SMD/0603	R11
1 kΩ/1 %/SMD/0603	R3-R5
2,2 kΩ/1 %/SMD/0603	R2, R7
10 kΩ/SMD/0603	R6, R9
PT10, liegend, 1 kΩ	R10

Kondensatoren:

4,7 nF/SMD/0603	C15, C16
100 nF/SMD/0603	C1, C3, C5, C8-C11
1 µF/SMD/0603	C2, C4, C6, C12-C14
10 µF/16 V	C7

Halbleiter:

ELV10978/SMD	IC1
74HCT174/SMD	IC2, IC3
TPS60400/SMD	IC4
74F00/SMD	IC5
BC848C	T1

IRLML6401/SMD

BCW67C/SMD	T2
LL4148	T3
LC-Display DM12232-05YT mit Beleuchtung	D1
	LCD1

Sonstiges:

Inkrementalgeber mit Taster, 12 Impulse/360°	DR1
Alu-Drehknopf mit Steckensatz, 28 mm	DR1
Mini-Drucktaster, B3F-4050, 1x ein	TA1, TA2
Tastkappen, 10 mm, Grau	TA1, TA2
Stiftleisten, 1x 2-polig, gerade, print	ST1-ST4
Stiftleiste, 1x 9-polig, gerade, print	LCD1
Sound-Transducer, 3 V, print	PZ1
4 Senkkopfschrauben, M2 x 14 mm, Kreuzschlitz	
4 Kunststoffschrauben, 2,5 x 12 mm	
4 Muttern, M2	
4 Fächerscheiben, M2	
4 Distanzrollen, M2 x 8 mm	
1 Gehäuse, komplett, bearbeitet und bedruckt	
4 Gehäusefüße, selbstklebend, 8 x 2,5 mm, schwarz	

Stückliste