



Wohlfühlen mit wakeboX – das einfache LINUX-Hausautomationssystem

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.

Ergänzend zur Beschreibung der Software im ersten Teil wird es im zweiten Teil ein universelles Bedien- und Anzeigeterminal geben – einen USB-HID-Client für den Nachtschrank, der über das veröffentlichte Schnittstellenprotokoll auch von eigenen Anwendungen angesprochen werden kann.

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-



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 Walz 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 alle 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 „Einmalschalter“ (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 ausgeben. 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: Absolutes Mini-Format – der SheevaPlug

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].

Zukünftig 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
```

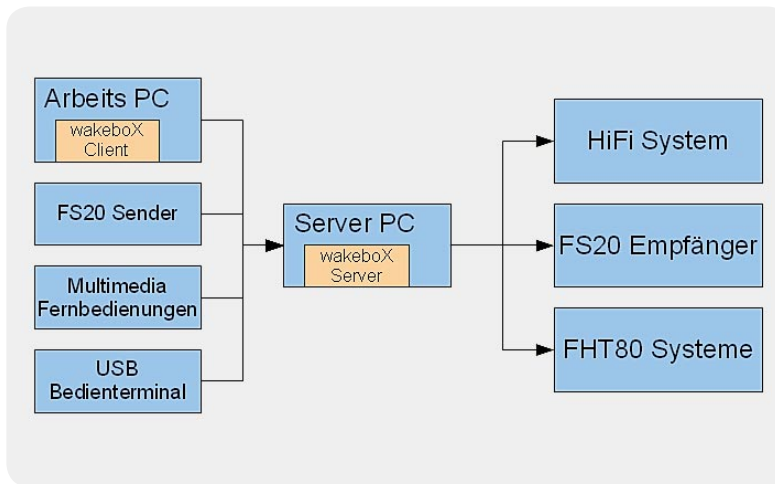


Bild 3: Die Systemübersicht über die Hardware des wakeboX-Systems

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“ abbrechen 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 Weltkugel (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 Weltkugel wird die Verbindung zum Server aufgebaut. Der Kontakt ist hergestellt, sobald die blaue „LED“ in der Weltkugel 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 wakeboX-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 Funksymbol („FS20-Aktoren“, Bild 8). Über das Lautsprechersymbol („Audio“) gelangt man zur Steuerung der Audioplayer (Bild 9), und die direkte Ablaufkontrolle ist über das Zahnrädersymbol („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 wakeboX, 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 wakeboX 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 („ream“) des FS20-Einmalschalters betätigt.

Sonnenauf- und -untergang

Ein weiteres Highlight für Aufweckabläufe ist natürlich der simulierte Sonnenaufgang. Dieser wird vom Dimmer-Aktor der wakeboX 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-Aktoren direkt steuerbar.



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



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



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://fedoraproject.org/de>
- [9] www.wakebox.de/download.html
- [10] www.mpg123.de
- [11] www.java.com/de/download/manual.jsp

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 Eingabebeziehung 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. Ein ähnliches Bedienteil von ELV wird in der nächsten Ausgabe des „ELVjournals“ vorgestellt. Viel Spaß und einen genussvollen Morgen mit wakeboX. **ELV**