

# DIY: Herstellung eines personalisierten Weckers mit Mikrocontroller und mp3-Player

Malte Ollenschläger

**Zusammenfassung**—Dieses Dokument beschreibt die Herstellung eines personalisierten Weckers. Das beinhaltet die Erstellung eines Gehäuses und das Schreiben der Software für den Mikrocontroller. Auch die Bauteilauswahl und die Fertigung von Platinen sind in diesem Projekt inbegriffen. Das Ziel ist einerseits die Fertigung des Weckers, andererseits die Durchführung eines Projektes mit verschiedenen Aufgabenbereichen, um das hierbei Gelernte, in folgenden Projekten zu einer effizienteren Arbeitsweise umzusetzen. Dies betrifft sowohl inhaltliche als auch formelle Arbeitsschritte.

## I. EINLEITUNG

Die Gesamtfunktion des Weckers setzt sich aus mehreren Teilfunktionen zusammen. Eine der grundlegenden Funktionen ist die *Anzeige der Uhrzeit* auf einem Display. Weiterhin wird das *Datum* durch die Zusammensetzung des aktuellen Monats als Wort und des Tages als Ordinalzahl angezeigt. Ein weiteres Symbol gibt an, ob der Alarm ein- oder ausgeschaltet ist. Das Umschalten funktioniert durch das Gedrückthalten eines Tasters, über den gleichzeitig bei kurzem Drücken die Displaybeleuchtung und die Beleuchtung der Bilder eingeschaltet werden kann. Die Implementierung der Weckfunktion sorgt bei eingeschaltetem Alarm dafür, dass zur konfigurierten Uhrzeit der mp3-Player gestartet und eine sich auf selbigem befindliche mp3-Datei abgespielt wird. Beim nächsten Weckruf wird die darauf folgende Datei abgespielt.

Über ein Menü, das über das Bedienen eines zweiten Tasters zu erreichen ist, sind folgende Elemente einstellbar:

- Uhrzeit
- Weckzeit
- Datum
- Lautstärke
- Boombox

Innerhalb des Menüs findet die Navigation über einen Drehencoder statt. Bei der Einstellung des Datums wird neben Tag und Monat auch das Jahr abgefragt. Dies dient der Erkennung von Schaltjahren, damit in diesen in der Datumsanzeige auf den 28. Februar statt dem 1. März der 29. Februar folgt. Mit dem Menüpunkt "Lautstärke" lässt sich eben diese des mp3-Players und damit die Lautstärke des Wecktons verändern. Wird der Menüpunkt "Boombox" ausgewählt, so kann über ein Cinch-Kabel eine externe Audio-Quelle angeschlossen werden und die darauf abgespielte Musik wird über die Lautsprecher des Weckers wiedergegeben. In dieser Zeit ist die Weckfunktion außer Kraft gesetzt.

Die Personalisierung des Weckers wird durch drei Eigenschaften erreicht. Die erste ist die Tatsache, dass die Weckmusik nach dem eigenen Geschmack angepasst werden

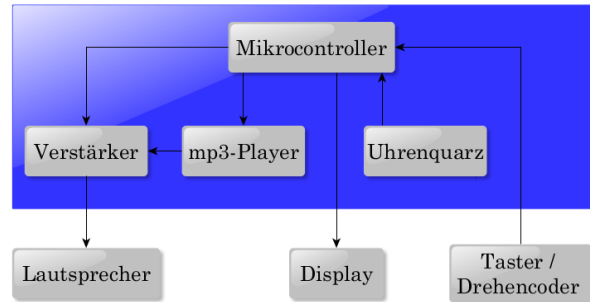


Abbildung 1: Hierarchie der Elementaren Bauteile des Weckers. Die Pfeilspitze zeigt in die Datenflussrichtung. Blau unterlegt sind die für den Benutzer nicht zugänglichen Bauteile im Gehäuse. Die unterste Zeile zeigt die Ein- und Ausgabebene, worüber der Benutzer mit dem Wecker interagiert.

kann, indem diese auf den mp3-Player kopiert wird. Die zweite personalisierte Eigenschaft ist, dass beispielsweise an Valentinstag und am Geburtstag der Person statt des Datums andere Bilder oder Texte eingeblendet werden. Der offensichtlichste Teil der Personalisierung liegt in der Herstellung des Gehäuses. In die Front des Weckers werden Motive zu denen die Benutzerin einen persönlichen Bezug hat, geschnitten. Diese werden zusätzlich zum Display beleuchtet, wenn der entsprechende Taster betätigt wird. Als Vorlage für dieses Projekt diente ein ebenfalls selbst hergestellter Wecker, der auf der Internetseite [www.wiesolator.de](http://www.wiesolator.de) vorgestellt wurde.

## II. HARDWARE

In diesem Abschnitt wird die genutzte Hardware erläutert. Hierzu wird in Teil II-A beschrieben, welche Bauteile benötigt werden. Der sich anschließende Teil II-C befasst sich mit der externen Beschaltung der Bauteile. In II-E wird auf die Planung und Fertigung der Platinen eingegangen.

### A. Bauteile

Der Schaltplan, der im Vorlageprojekt genutzt wurde, dient als Anhaltspunkt für die Hardware, die benötigt wird. Daraus wurde das Schaubild in Abb. 1 erstellt, um eine Übersicht der wichtigsten Bauteile zu erhalten. Eine Auflistung aller benötigten Bauteile ist in Tabelle I aufgeführt. Die benötigten Widerstände und Kondensatoren sind nicht aufgeführt, da diese den Datenblättern der jeweiligen Bauteile zu entnehmen sind. Zusätzlich werden für das Entprellen der Taster und des Drehencoders 100nF Kondensatoren eingesetzt.

Funktionseinheit	Bezeichnung
Buchsenleiste	EBLG-20-1-X
Display	EA DOGL128W-6
Displaybeleuchtung	EA LED68x51-W
Drehencoder	STEC11B
Lautsprecher	Visaton FR8-8
Mikrocontroller	Atmega328P
mp3-Player	Intenso Music Walker
Schaltnetzteil 5V/1A	goobay 54805
Spannungsregler 1,5VDC	LM317
Spannungsregler 3,3VDC	LM3940
Spanplatten	
Schrauben	
Muttern	
Taster	FEHLT
Uhrenquarz 32kHz	-
Verstärker	TDA7266

Tabelle I: Benötigte Bauteile. Notwendige Kondensatoren und Widerstände sind nicht angegeben. Diese sind in den jeweiligen Datenblättern zu finden.

Neben dieser Hardware werden Spanplatten zur Fertigung des Gehäuses, sowie einige Schrauben benötigt.

### B. Gehäuse

In die vier Ecken der Platine werden Löcher gebohrt, ebenso im gleichen Abstandsmaß in die Front des Weckers. Anschließend kann die Platine per Schraube und Mutter von innen gegen das Gehäuse gedrückt werden, sodass das Display fixiert ist und durch die rechteckige Öffnung in der Front des Gehäuses betrachtet werden kann.

### C. Dimensionierung

Grundsätzlich wurden die in den Datenblättern empfohlenen Kondensatoren und Widerstände eingesetzt. Zum Entprellen der Taster sind in den entsprechenden Datenblättern keine Informationen vorhanden. Es werden Keramik Kondensatoren der Größe  $C = 100nF$  genutzt. Bei dem Spannungsregler für den mp3-Player (LM317) muss eine individuelle Dimensionierung vorgenommen werden, da der Ausgang des gewählten Spannungsreglers variable Spannungen erzeugen kann. Die benötigte Versorgungsspannung beträgt 1,5V. Nach der im zugehörigen Datenblatt [QUELLE] angegebenen Formel, kann für eine gewünschte Ausgangsspannung die Größe der benötigten Widerstände berechnet werden. Somit ergibt sich die Größe von  $R_2$  nach folgender Gleichung:

$$R_2 = \left( \frac{V_{out}}{V_{ref}} - 1 \right) / \left( \frac{1}{R_1} + I_{adj} \right) \quad (1)$$

Für den Ausgleichsstrom wurden  $50\mu A$  und damit die Hälfte des Maximums angenommen. Dieser wird vom Spannungsregler so geregelt, dass  $V_{ref} = 1,25V$  gilt. Bei einer Wahl von  $R_1 = 10k\Omega$  ergibt sich für  $R_2 \approx 1,3k\Omega$ . Zur Beleuchtung der beiden Bilder auf der Frontseite des Weckers wird eine Transistorschaltung (??) benötigt. Diese wird durch ein Kontrollsignal vom Mikrocontroller, welches an die Basis

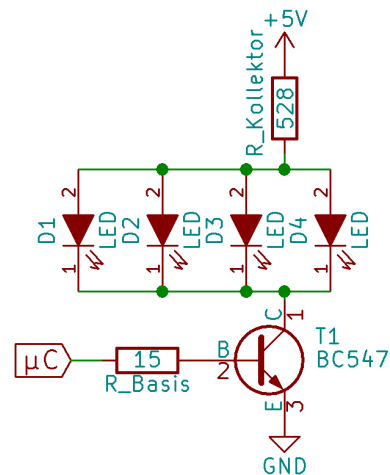


Abbildung 2: Transistorschaltung zur Ansteuerung von LEDs

des Transistors angeschlossen ist, gesteuert. Ein BC547C Transistor schaltet 4 LEDs mit je 20 mA. Die Berechnung der Widerstände ergibt sich aus folgenden Formeln:

$$R_{Basis} \leq \frac{U_{\mu C} - U_{BE}}{I_B} = \frac{3,3V - 0,66V}{5mA} \approx 528\Omega \quad (2)$$

$$R_{Kollektor} = \frac{U_{cc} - U_{LED} - U_{CE}}{I_C} = \frac{5V - 3,6V - 0,2V}{80mA} \approx 15\Omega \quad (3)$$

Da der Mikrocontroller eine Versorgungsspannung von 3,3V hat, ist das Steuersignal  $U_{\mu C}$  ebenso groß. Die Basis-Emitter- und Kollektor-Emitter-Spannung sowie der Basisstrom sind dem Datenblatt des BC547C entnommen. Gleiches gilt für die an den LEDs abfallende Spannung. Nach den Berechnungen in (2) und (3) wurde der Basiswiderstand auf  $440\Omega$  und der Kollektorwiderstand auf  $15\Omega$  gewählt. Der Kollektorstrom von  $I_C = 80mA$  ist das Vierfache des Vorwärtstroms einer LED ( $I_F = 20mA$ ), da er sich nach der ersten Kirchhoff'schen Regel gleichmäßig auf die vier LEDs aufteilt.

### D. Gehäuse

### E. Platinen

Die für dieses Projekt benötigten Platinen wurden mit KiCAD geplant und anschließend im FabLab der Friedrich-Alexander-Universität Erlangen-Nürnberg erstellt. Die erste Platine konnte nicht verwendet werden, da das ausgedruckte Layout nicht den tatsächlichen Dimensionen entsprach. Die Ursache hierfür liegt vermutlich im Export des Schaltplans in eine PDF-Datei. Korrekte Maße bleiben nur Erhalten, wenn in eine PostScript-Datei anstatt in ein PDF exportiert wird.

Eine Platine ist für das Display und die zugehörigen Bauelemente (Kondensatoren, Widerstände) ausgelegt. Über einen Steckverbinder kann diese Platine mit einer zweiten verbunden werden, welche die anderen Bauelemente wie zum Beispiel den Mikrocontroller und den Audio-Verstärker enthält.

Für die Displayplatine wurden die Anschlüsse des Displays nicht direkt auf der Platine verlötet. Es wurden Buchsenleisten

verlötet in welche das Display samt verlöteter Hintergrundbeleuchtung eingesteckt wird. Dies dient einerseits dazu, dass das Display zum Wechsel oder zur Reparatur ausgebaut werden kann; andererseits dazu, dass der Abstand zwischen Gehäus-einnenseite und Platine vergrößert wird. Dies ist notwendig, damit der Drehencoder, welcher auf der Platine befestigt wird, nicht zu weit nach vorne aus dem Gehäuse herausragt. Außerdem dient es der Herstellung einer kraftschlüssigen Verbindung zwischen Display und Gehäuse (siehe II-B). Für die Erstellung der Platine wurde eine Leiterbahnbreite von  $0,5\text{mm}$  gewählt, da genug Fläche vorhanden war.

Für die Hauptplatine wurde ebenfalls eine Leiterbahnbreite von  $0,5\text{mm}$  gewählt. Neben der Displayplatine wurden weitere Bauelemente per Steckverbinder mit der Hauptplatine verbunden. Diese sind:

- Lautsprecher
- Taster
- Drehencoder
- LED-Beleuchtung (Bilder)
- mp3-Player

Bei der Erstellung des PCB-Layouts wurde nicht durchgängig die Anordnung der Bauteile auf dem Schaltplan beachtet. Eine Vorsortierung in Bauteilgruppen erleichtert das Routing und ist somit ratsam.

#### F. Pinbelegung Mikrocontroller

Da es für den Oszillator keine Auswahlmöglichkeiten bezüglich der Pinbelegung gibt, wurden ihm als erstes die entsprechenden Pins am Mikrocontroller zugewiesen. Die Pins für den Anschluss des Displays wurden so gewählt, dass alle nebeneinander liegen, damit das Layout der Platine erleichtert wird und lediglich gerade Leiterbahnen zwischen den Pins des Mikrocontrollers und dem betreffenden Steckverbinder erstellt werden müssen. Gleiches gilt für die Belegung der Pins zur Ansteuerung des mp3-Players.

Die Auswahl für die Belegung der Taster unterliegt lediglich der Einschränkung, dass die Taster (Licht und Drehencoder) einem anderen Interrupt zugeordnet werden als eine Drehung des Drehencoders. Dies führt zu einer vereinfachten Auswertung der Drehrichtung, siehe III-B.

#### G. Ansteuerung mp3-Player

Um ein Bild davon zu bekommen, wie die Steuerung des mp3-Players funktioniert, wurde dieser zunächst aus seinem Gehäuse ausgebaut. Durch Testen wurde die Funktionsweise der benötigten Taster simuliert. In Abb. ?? ist eine Zeichnung des mp3-Players dargestellt, die zum näheren Verständnis des folgenden Textes herangezogen werden kann.

Der silberne Steuerknopf hat auf zwei gegenüberliegenden Seiten je 3 Kontakte. Darüber liegt rings um diesen Knopf unterhalb der Plastikabdeckung eine Massefläche. Wird der Knopf in eine bestimmte Richtung bewegt, so werden zwei der 6 Pins auf Masse gezogen und daraus resultiert ein Steuerungssignal. Welches Ereignis durch die Kombination der verschiedenen Pins ausgelöst wird, ist in Tabelle II dokumentiert. Zur Betätigung des Ein- und Ausschalers, der bei kurzem Drücken

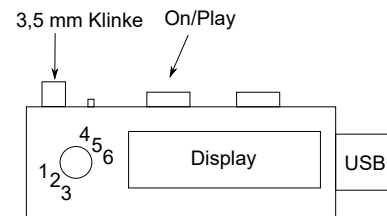


Abbildung 3: Transistorschaltung zur Ansteuerung von LEDs

für Play/Pause genutzt wird, muss eine der vier Lötstellen auf das Level  $1,5\text{V}$  gehoben werden. Der Taster weist vier Lötstellen auf. Davon befinden sich zwei auf Oberseite und zwei in Richtung Mitte der Platine. Von letzteren muss die Lötstelle, welche in Richtung der  $3,5\text{mm}$ -Klinken-Buchse liegt, auf  $1,5\text{V}$  angehoben werden. Um das  $3,3\text{V}$  Schaltsignal des Mikrocontrollers auf  $1,5\text{V}$  für den mp3-Player zu verringern, wurden 3 Dioden vom Typ 1N4448 in Reihe zwischen den Ausgang des Mikrocontrollers und den betreffenden Konnektor zum mp3-Player geschaltet. Über den Dioden fällt jeweils etwa  $0,6\text{V}$  ab, sodass  $3,3\text{V} - 3 \cdot 0,6\text{V} = 1,5\text{V}$  an dem Taster anliegen, wenn der Pin des Mikrocontrollers auf  $3,3\text{V}$  gesetzt ist.

Pin A	Pin B	Level	Ereignis
1	5	0V	Song wechseln
3	5	0V	Lautstärke -
4	5	0V	Lautstärke +
On-1	-	$1,5\text{V}$	Ein/Aus, Play/Pause

Tabelle II: Tastenkombination für Auslösung verschiedener Ereignisse

### III. SOFTWARE

Um die gewünschten Inhalte auf dem LCD-Display anzuzeigen, müssen diese Elemente zunächst generiert werden. Dies geschah mittels Inkscape. Anschließend wurden diese Inhalte einen selbst geschriebenen Java-Programm zugeführt, welches in Abschnitt III-A beschrieben wird. Der zweite Abschnitt befasst sich mit dem umfangreicheren Code, der auf dem Mikrocontroller eingesetzt wird.

#### A. Bildkonversion

Für die Entwicklung eines Bildkonverters in Java wurde Eclipse genutzt. Ziel ist die Umwandlung von grafischen Dateien in Binärcode, also ein monochrom Bild. Außerdem sollen jeweils 8 Pixel zu einer Hexadezimalzahl zusammengefasst werden.

Zunächst werden Dateien mit einzelnen Buchstaben, Symbolen und Bildern in Inkscape erstellt. In Schriftgröße 18 gibt ein zentrierter Großbuchstabe *J* die vertikale Position aller weiteren Buchstaben vor, da dies der höchste Buchstabe ist. Die erstellte Grafik wird als \*.png exportiert und dem Java-Programm zugeführt. Dieses trägt den Namen *Bildkonverter*. Wird es geöffnet, sind drei Buttons und ein Eingabefeld zu sehen. Über *Datei wählen* wird das zu verarbeitende Bild

ausgewählt. Die Umwandlung startet durch Klicken des entsprechenden Buttons. Zunächst wird durch Auswertung des Rotkanals jedes Pixels ein Binärwert erzeugt. Ist der Wert des Rotkanals größer oder gleich der angegebene Schwellenwert, so wird das entsprechende Pixel gesetzt. Ist es geringer als der Schwellenwert, wird dieser Pixel auf null gesetzt. Anschließend werden jeweils 8 aufeinander folgende Pixel einer Spalte zu einem Hexadezimalwert zusammengefasst, um das Handling zu erleichtern. Wenn ein Bild eine Höhe von 16px hat und eine Breite von 3px, so werden 3 Mal zwei Hexadezimalwerte erzeugt. Diese sechs Werte werden in geschweiften Klammern ausgegeben, damit diese in einem Alphabet-Array im Code für den Mikrocontroller gespeichert werden können. Weiterhin wird eine Vorschau des Binärbildes angezeigt. So müssen die Daten nicht erst auf den Mikrocontroller überspielt und am LCD-Display angezeigt werden, um Fehler zu erkennen. Diese können beispielsweise durch Änderung des Schwellenwertes oder durch das Setzen oder Löschen einzelner Pixel in dem Bild mittels Bildbearbeitungssoftware (zum Beispiel Paint) korrigiert werden. Zum Beenden des Konverters dient der Button *Schließen*. Die Ausgabe befindet sich nun in einer Datei mit dem Namen *Konverter-output.txt* in dem Ordner, in dem das Java-Programm gestartet wurde.

## B. Atmel-Code

Der Code für den Mikrocontroller setzt sich aus verschiedenen Dateien zusammen. Die Hauptaufgabe übernimmt eine Endlosschleife in der Main-Funktion der Datei *EADOGL128.c*. In *EADOGL128\_commands.c* wurden die Befehle zur Ansteuerung des LCD-Displays implementiert. Um den mp3-Player und den Audio-Verstärker anzusprechen, werden die in *mp3.c* erstellten Funktionen genutzt. Nicht selbst geschriebener Code wurde in Form von Standardbibliotheken wie *avr/io.h* genutzt sowie in der Datei *sbit.h* [QUELLE], welche es ermöglicht, Pins wie Variablen ansprechen zu können.

*EADOGL128.c*: Es gibt mehrere Interrupt-Service-Routinen, die Einfluss auf die ausgeführten Aktionen in der Main-Funktion nehmen. Im Normalzustand befindet sich der Mikrocontroller im sleep-Modus. Um in diesen einzutreten sind einige Vorbereitungen notwendig, die in der Funktion *goodNight* implementiert sind. Das Verlassen des sleep-Modus geschieht durch eines von mehreren möglichen Interrupts. Hierzu zählen die Bedienung des Drehencoders und des Tasters sowie ein Timer-Overflow aufgrund der Zeitmessung mittels Uhrenquarz. Dieser Overflow findet ein Mal pro Sekunde statt. Wenn sich auf dem LCD-Display durch das Zeit-Update nichts ändern muss ( $\text{Sekunde} < 60$ ), wird der sleep-Modus erneut aktiviert.

Im Falle einer Benutzereingabe oder eines notwendigen Updates des LCD-Displays, wird die Funktion *goodNight* verlassen und das betreffende Ereignis wird abgearbeitet. Danach tritt der Mikrocontroller erneut in den sleep-Modus ein.

Interrupt-Service Routinen:

Die ISR *PCINT0\_vect* setzt bei gedrückten Tasten das entsprechende Flag auf *high*. Somit wird zum Beispiel das Einschalten des Lichts in der main-Funktion eingeleitet. Würde die Drehung des Drehencoders im gleichen Interrupt erfasst

werden, so müsste der letzte bekannte Status des Drehencoders gespeichert werden, damit überprüft werden kann, ob eine Änderung stattgefunden hat, oder lediglich ein Taster bedient wurde. Durch die Zuweisung der Signale, die bei einer Drehung verändert werden an eine andere ISR (*PCINT2\_vect*), als jener die bei Tastendruck aufgerufen wird (*PCINT0\_vect*), kann dieser Schritt eingespart werden. Die ISR (*PCINT2\_vect*) vergleicht den Status der beiden betreffenden Signale. Die Drehrichtung lässt sich daraus ermitteln, ob beide Signale gleich (beide *high* oder beide *low*) oder unterschiedlich sind. Je nach Ergebnis dieses Vergleichs wird bei einer detektierten Rechtsdrehung eine Zählvariable inkrementiert sowie bei Linksdrehung dekrementiert. Diese muss bei Abruf auf *null* zurückgesetzt werden.

Bei eingeschalteter Beleuchtung bewirken alle Benutzereingaben eine Verlängerung der Leuchtdauer so, als ob der Taster für die Beleuchtung zu diesem Zeitpunkt betätigt wurde.

Die dritte ISR wird aufgerufen, wenn ein Overflow in dem vom Uhrenquarz getakteten Counter stattfindet. Da ein 8-bit Timer mit einem Prescaler von 128 zum Inkrementieren des Counters genutzt wird, findet dieser Overflow ein Mal pro Sekunde statt. Hierbei wird die Variable *sekunde* um eins inkrementiert. Wenn diese Variable den Wert 59 übersteigt, wird ein Flag gesetzt, welches das Update der Minute und gegebenenfalls der Stunde sowie des Datums in der main-Funktion einleitet. Im gleichen Zug wird damit der Inhalt des LCD-Displays entsprechend verändert.

Eine vierte ISR wird genutzt um die Dauer eines Tastendrucks zu ermitteln. Bei langem Drücken des Drehencoders wird so das Menü aufgerufen. So ist es möglich, dass der Taster für das Licht bei kurzem Drücken zum Einschalten der Beleuchtung und bei langem Drücken zum Ein-/Ausschalten des Alarms führt.

## IV. BEDIENUNG

### A. Alarm

Um den Alarm ein- oder auszuschalten wird der Taster für mindestens 1,5s gedrückt. Ausschalten des Alarms sowie stoppen des Weckrufs geschieht durch Betätigung des Tasters von 1,5s oder mehr. Während des Weckrufs ist es außerdem möglich, in den snooze-Modus zu wechseln. Dies geschieht durch kurzes Drücken des Tasters oder Drehen des Drehencoders. Ein erneuter Weckeruf wird nach der im Menü eingestellten Snoozedauer (in Minuten) gestartet. Erneutes snoozen ist nicht möglich und der Weckruf muss über ein 1,5-sekundiges Drücken des Tasters ausgeschaltet werden.

### B. Menü

Das Menü kann über ein langes Drücken des Drehencoders aufgerufen werden. Navigiert wird über die Drehung desselben. Soll ein Punkt im Menü aufgerufen wird, so muss die Auswahl durch kurzes Drücken auf den Drehencoder getätigt werden. In Untermenüs wie beispielsweise der Uhrzeiteinstellung wird ebenfalls durch eine Drehung navigiert. Soll zum Beispiel die Stunde verändert werden, muss die Markierung über das entsprechende Element navigiert werden. Nach einer Bestätigung durch erneut kurzes Drücken ändert sich das

Navigationssymbol und die Stunde der Uhrzeit kann verändert werden. Durch erneutes Drücken des Drehencoders wird dieser Modus verlassen. Das Untermenü kann über den angezeigten Haken verlassen werden.

### *C. Lautstärke*

Für die Einstellung der Lautstärke des Weckrufs muss der entsprechende Unterpunkt im Menü aufgerufen werden. Daraufhin werden Verstärker und mp3-Player eingeschaltet. Letzterer benötigt etwa zehn Sekunden. Während dieser Zeit erscheint der Schriftzug „Bitte warten“. Die Musik geht an und die Lautstärke kann mittels Drehencoder verstellt werden.

### *D. Boombox*

Bei Auswahl des Menüpunktes Boombox, wird der Verstärker eingeschaltet. Nun kann über die auf der Platinen befindlichen Cinch-Buchsen und ein entsprechendes Kabel (z.B. 3,5mm-Klinke ↔ Cinch) eine externe Audioquelle wie ein Handy oder Laptop angeschlossen werden. Dessen Audiosignal wird dann über die Lautsprecher wiedergegeben.

## V. QUELLEN

<http://www.avrfreaks.net/sites/default/files/SBIT.H>