BambiControl: Bauanleitung, Konfigurationen, Beschaffung, Einsatz

Version 0.9.6

Inhalt

1 Zu diesem Handbuch ... 1
2 Haftung und andere gesetzliche Hinweise ... 1
3 Einleitung ... 2
4 Hardwarekomponenten und Beschaffung ... 3
5 Stromverbrauch ... 4
6 Bauanleitung ... 5
6.1 Verbindungen im Überblick ... 5
6.2 Layoutplan für Europlatine ... 6
6.3 Löten Schritt für Schritt ... 8
6.3.1 Stiftleiste an Aafruit Trinket anlöten ... 8
6.3.2 Vorbereiten der Europlatine ... 9
6.3.3 Einlöten des Adafruit Trinket ... 9
6.3.4 Verlöten der Leiterbahnen ... 10
6.3.5 Verlöten des BC337 ... 11
6.3.6 Anlöten von zwei kurzen Kabeln an den Piezosignalgeber ... 11
6.3.7 Anlöten von zwei kurzen Kabeln an den Fotowiderstand ... 11
6.3.8 Zusätzliche Modifikationen zur Laufzeitverlängerung ... 12
6.4 Einbau in Gehäuse ... 13
6.4.1 Bohrung für Piezosignalgeber ... 13
6.4.2 Einbau Fotowiderstand ... 14
6.4.3 Einbau Platine ... 14
6.4.4 Bilder vom ersten Prototypen ... 14
6.5 Software aufspielen ... 15
6.5.1 Adafruit Trinket Treiber herunterladen und installieren ... 15
6.5.2 Anschluss des Adafruit Trinket ... 15
6.5.3 Programmier-Software herunterladen ... 16
6.5.4 Programmierung ... 16
7 Betrieb und Einsatz ... 21
7.1 Inbetriebnahme ... 21
7.2 Arbeitsweise ... 21
7.2.1 Signalphase ... 22
7.2.2 Ruhephase ... 22
7.3 Wirkungsbereich ... 22

Versionen dieses Dokuments

Version / Datum	Anmerkungen
Version 0.9 vom 03.06.2016	Initialversion
Version 0.9.1 vom 10.06.2016	Fehler im Schaltbild behoben
Version 0.9.2 vom 10.06.2016	Neue Programmiersoftware dokumentiert
Version 0.9.5 vom 12.06.2016	Neues Platinenlayout
Version 0.9.6 vom 16.06.2016	Längere Pausen zwischen den Signalen

Verzeichnis der Abbildungen

Abbildung 1: Europlatine BambiControl Oberseite farbig 6
Abbildung 2: Europlatine BambiControl Oberseite schwarzweiß 6
Abbildung 3: Europlatine BambiControl Unterseite farbig 7
Abbildung 4: Europlatine BambiControl Unterseite schwarzweiß 7
Abbildung 5: Oberseite Adafruit Trinket (zeigt 5V-Version des Trinket, zum Einsatz muss die 3,3 V-Version kommen) 8
Abbildung 6: Adafruit Trinket mit Beinchen mit Größenvergleich 9
Abbildung 7: PINS zur Fixierung des Adafruit Trinket 10
Abbildung 8: Europlatine mit gelöteten Leiterbahnen (es geht auch schöner) 10
Abbildung 9: Anlöten Piezosignalgeber an Platine 11
Abbildung 10: Anlöten LDR an Platine 12
Abbildung 11: Adafruit Trinket LEDs 13
Abbildung 12: Erster Prototyp offen 14
Abbildung 13: Erster Prototyp Unterseite 15
Abbildung 14: Startbildschirm der Programmiersoftware 16
Abbildung 15: Programmiersoftware: Nächster Schritt 17
Abbildung 16: Adafruit Trinket Programmierknopf (roter Kreis) 17
Abbildung 17: Ende der Programmierung 18
Abbildung 18: Fehlerbeispiel 1: Adafruit Trinket nicht gefunden 19
Abbildung 19: Fehlerbeispiel 2: USB 3.0 Port 19
Abbildung 20: Erfolgreiche Programmierung 20

1 Zu diesem Handbuch

Dieses Handbuch befindet sich, genauso wie die Hardwarelösung in der permanenten Weiterentwicklung. So kann es durchaus sein, dass dieses Handbuch nicht ganz der aktuellen Hardwareempfehlung folgen kann – und umgekehrt, dass manche Funktionen in diesem Handbuch bereits beschrieben sind, aber noch nicht vollständig implementiert und/oder getestet.

Dafür bitte ich um Verständnis, entwickle ich BambiControl bislang völlig allein und diese Vorgehensweise erspart mir einiges an Zusatzarbeit.

2 Haftung und andere gesetzliche Hinweise

Diese Anleitung wird dem Verwender unentgeltlich im Rahmen eines Gefälligkeitsverhältnisses zur Verfügung gestellt; eine vertragliche Verbindung wird damit nicht begründet. Die Verwendung der Anleitung erfolgt auf eigenes Risiko; eine Haftung des Zurverfügungstellenden wird nicht begründet.
Es wird somit jedwede Haftung bei Einsatz der Soft- und/oder Hardwarekomponenten von BambiControl abgelehnt. Dies beinhaltet auch Haftung für gesundheitliche Schäden durch die Lautstärke der eingesetzten Komponenten.

Ebenfalls wird jede Haftung für etwaige Empfehlungen zu Hard- und Softwarebeschaffungen, Einstellungsvorschlägen, Einsatz des Systems o.ä. abgelehnt.

Der Nutzer dieser Anleitung erklärt sich hiermit einverstanden.

3 Einleitung

BambiControl ist eine Anlage zur Vergrämung von Wild von Flächen mittels optischer und akustischer Signale. Der Einsatz scheint denkbar, um Rehwild von Flächen zu vergrämen, beispielsweise um Geißen dazu zu bewegen, ihre Kitze von Wiesen zu holen, bevor diese gemäht werden. Auch zur Verhütung von Schwarzwildschäden kann es eingesetzt werden. Es kommen hierbei Standardelektronikkomponenten zum Einsatz die problemlos bei diversen Internetversendern bestellt werden können.

Zentrale Elemente von BambiControl sind:

- Ein Mikrocontroller zur Steuerung der Signale: Adafruit Trinket **in der 3,3 V Version (!)**: https://www.adafruit.com/products/1500
- Ein Sensor für die tageslichtabhängige Steuerung, konkret ein Fotowiderstand, Modell LDR 07
- Ein 12 V Piezosignalgeber mit oder ohne Lichtsignal
  WARNUNG: Der vorgeschlagene Piezosignalgeber ist mit 95 dB sehr laut. Wenn Sie das Gerät testen, sollten Sie die Nutzung eines Gehörschutzes in Erwägung ziehen!
- Eine 12 V Batterie (Bleigelakku oder ausgemusterte Autobatterie)
- Ein NPN-Transistor Model BC 337 (getestet wurde mit einem BC337-40, andere Varianten wie BC337-25 sollten ebenfalls funktionieren)
- Eine Software, die auf den Adafruit Trinket aufgespielt werden muss und von www.trapcontrol.de/downloads heruntergeladen werden kann

4 Hardwarekomponenten und Beschaffung

Die oben beschriebenen Komponenten können bei den folgenden Bezugsquellen bestellt werden:

Komponente	Bezugsquelle	Preis ohne Versand
Adafruit Trinket 3,3 V	http://www.exp-tech.de/mainboards/arduino/adafruit-trinket-mini-microcontroller-3-3v-logic **ACHTUNG: Achten Sie darauf, die 3,3 V-Version zu bestellen!**	6,75 €
Fotowiderstand LDR 07	https://www.reichelt.de/Fotodioden-etc-/LDR-07/3/index.html?ACTION=3&GROUPID=3045&ARTICLE=10208	1,25 €
12 V Piezosignalgeber mit oder ohne Lichtsignal	https://www.conrad.de/de/piezo-summer-geraeusch-entwicklung-95-db-12-vdc-inhalt-1-st-1164631.html **WARNUNG: Der vorgeschlagene Piezosignalgeber ist mit 95 dB sehr laut. Wenn Sie das Gerät testen, sollten Sie die Nutzung eines Gehörschutzes in Erwägung ziehen!**	5,09 €
NPN-Transistor Model BC 337	https://www.conrad.de/de/bipolar-standard-leistungstransistor-fairchild-semiconductor-bc33725ta-npn-gehaeuseart-to-92-i-c-1-a-emitter-sperrspannung-u-ceo-45-v-154598.html	0,20 €
Gehäuse zur Aufnahme der Elektronik	https://www.conrad.de/de/abb-2tka140012g1-abzweigkasten-86-x-86-weiss-weiss-ip65-551335.html	2,49 €
Batterie 12 V	Beliebig, 1 Ah ergibt eine Laufzeit von ca. 5 Tagen (noch nicht abschließend getestet): - https://www.amazon.de/MULTIPOWER-MP1-2-12-BLEIAKKU-BATTERIE-1-2Ah/dp/B0071OTXAG/ref=sr_1_1?ie=UTF8&qid=1465328599&sr=8-1&keywords=bleigelakku+12V - https://www.akkuline.de/multipower-bleiakku-mp2-9-12-12-0volt-2-9ah-mit-4-8mm-steckanschluessen/3638_item.aspx	Ab 7.90 €
Summe:		ab ca. 23,00 € (zzgl. Versand)

5 Stromverbrauch

Im Wesentlichen sind die Hauptstromverbraucher

- der Adafruit Trinket und der
- Piezosignalgeber

Der Adafruit Trinket verbraucht im Betrieb (also in den Signalphasen) ohne Modifikationen (s.u.) ca. 12 mA, sofern man ihn modifiziert (Auslöten der grünen LED, s.u.) liegt er bei ca. 9 mA. In den Ruhephasen liegt der Stromverbrauch bei unter 1 mA.
Der Piezosignalgeber verbraucht in der Nutzung (laut Hersteller) 18 mA.

Gerät	Signalphase	Ruhephase
Adafruit Trinket	9-12 mA	1 mA
Piezosignalgeber	18 mA	0 mA

Jede Signalphase dauert zwischen 30 und 180 Sekunden. Zu vereinfachten Berechnung gehen wir von einem Mittelwert aus von 105 Sekunden. Jede Ruhephase dauert zwischen 300 und 600 Sekunden. Zu vereinfachten Berechnung gehen wir von einem Mittelwert aus von 450 Sekunden. [Anmerkung: Zwischenzeitlich sind die Ruhephasen länger, also die Laufzeit eher auch].
Wir nehmen an, dass ein typischer Betriebstag bei 12 Stunden absoluter Ruhephase und 12 Stunden mit Wechsel zwischen Signal- und Ruhephasen (beispielweise von 20:00 bis 08:00 Uhr) liegt. Mit den oben genannten Mittelwerten ergeben sich ca. 77 Ruhe- und ca. 77 Signalphasen. In den Signalphasen wiederum wird der Piezosignalgeber auch nicht permanent genutzt, im Mittel in der Hälfte der Zeit.
Es ergibt sich folgender Stromverbrauch pro Tag:

Phasen	Ohne Modifikationen	Mit Modifikationen
12 h Ruhephase		
Adafruit Trinket	12 mA	9 mA
12 h Wechsel Ruhe / Signalphase mit 77 Ruhe-/ Signalphasen		
Adafruit Trinket	27 mA	20 mA
Piezosignalgeber	175 mA	175 mA
Summe bei 12 Stunden Aktivität:	224 mA	217 mA
Summe bei 8 Stunden Aktivität:	157 mA	152 mA

Das ist konservativ gerechnet, da hier angenommen wird, dass der Piezosignalgeber einen Dauerton erzeugt, durch die Unterbrechungen ist der Stromverbrauch deutlich niedriger.
Im Ergebnis kann man davon ausgehen, dass ein 1 Ah Akku eine Laufzeit von ca. 5 Tagen hat.

6 Bauanleitung

6.1 Verbindungen im Überblick

Es sind nur wenige Verdrahtungen erforderlich:

- Batterie 12 V (Plus) an
    - Trinket „BAT“
    - Piezosignalgeber „PLUS“ / „ROT“
- Batterie GND (Minus)
    - an Trinket „GND“
    - an das linke Bein des BC 337 (von der flachen Seite gesehen)
- Trinket „#2“ an das mittlere Bein des BC 337
- Trinket „#3 “ an das eine Bein des Fotowiderstandes
- Trinket „GND“ an das andere Bein des Fotowiderstandes

Man kann die Verdrahtungen auch „on the Fly“ machen, besser und haltbarer ist die Nutzung einer Europlatine.

6.2 Layoutplan für Europlatine

Nachfolgend ein Vorschlag eines Layoutplan für eine Europlatine der Größe 35 x 30 (BxH) mm.

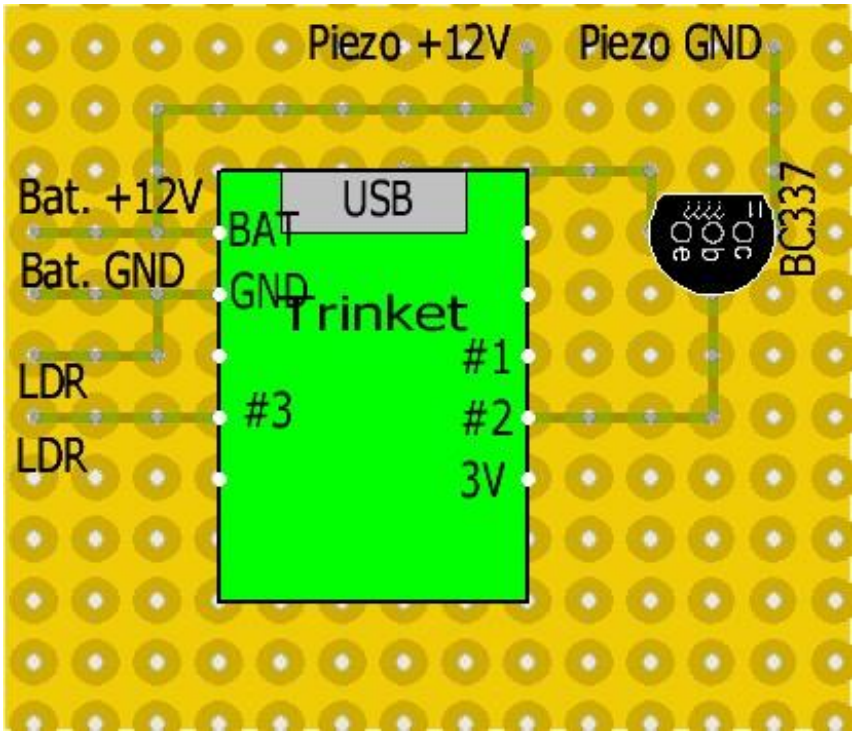

Abbildung 1: Europlatine BambiControl Oberseite farbig

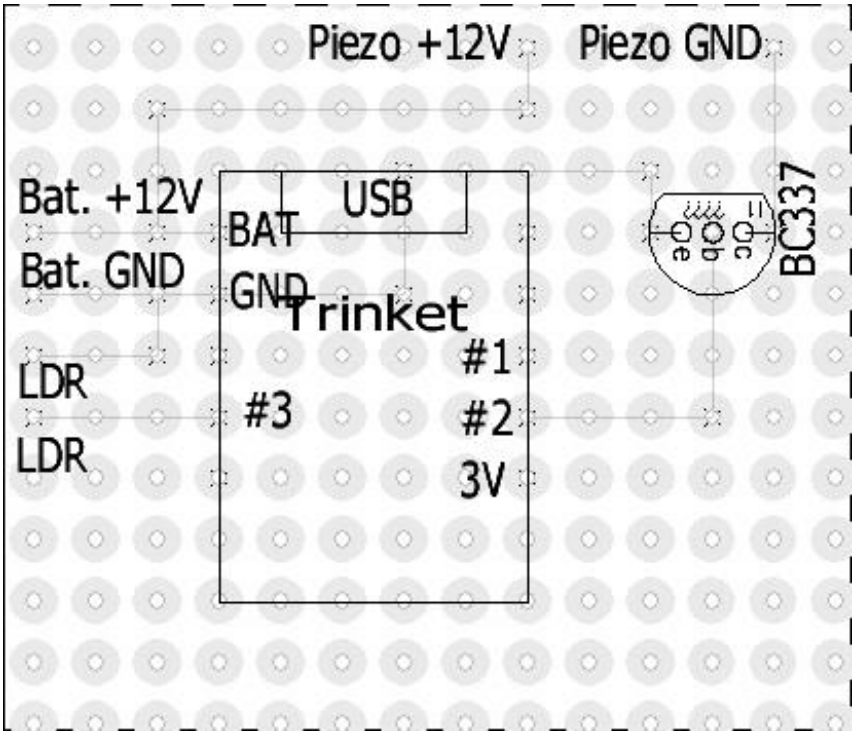

Abbildung 2: Europlatine BambiControl Oberseite schwarzweiß

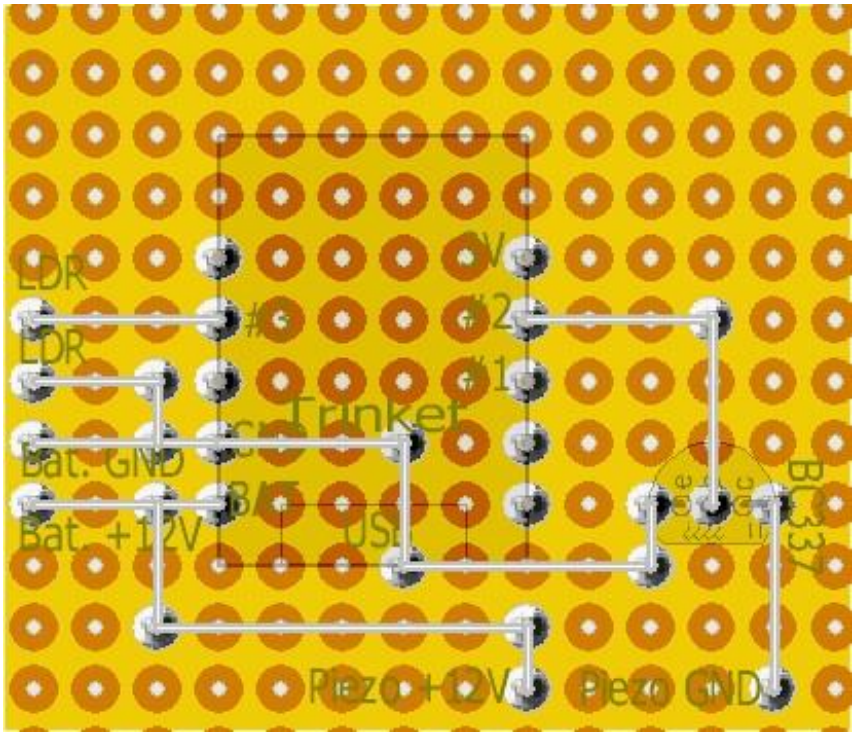

Abbildung 3: Europlatine BambiControl Unterseite farbig

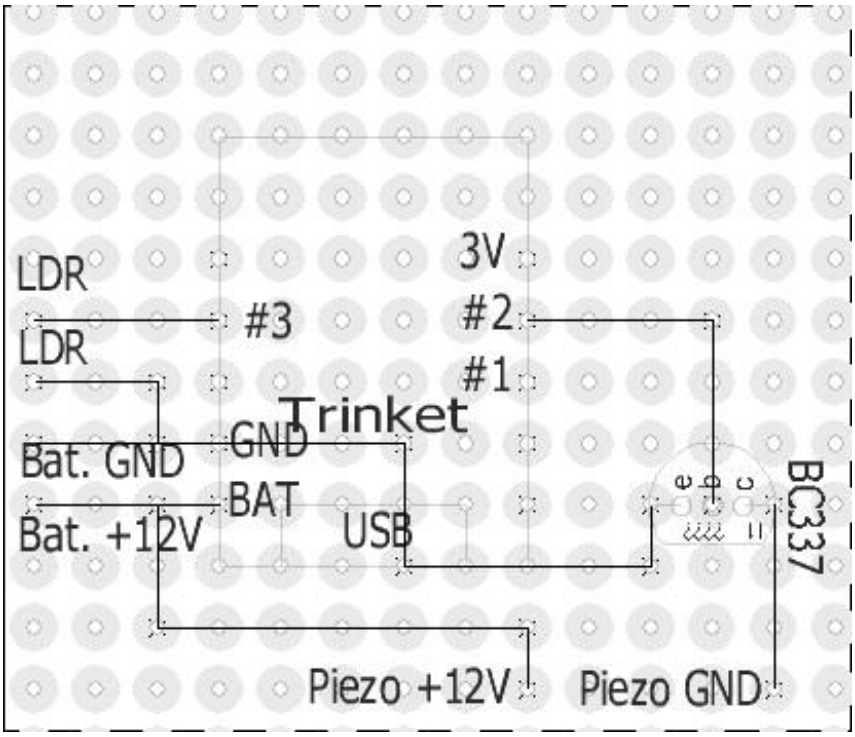

Abbildung 4: Europlatine BambiControl Unterseite schwarzweiß

6.3 Löten Schritt für Schritt

6.3.1 Stiftleiste an Aafruit Trinket anlöten

Bei der Lieferung des Adafruit Trinket ist eine lange Stiftleiste dabei. Hiervon brechen wir 2 Stücke à 5 Pins ab und löten diese an den Adafruit Trinket an, so dass (wichtig !!!) diese Seite oben ist:

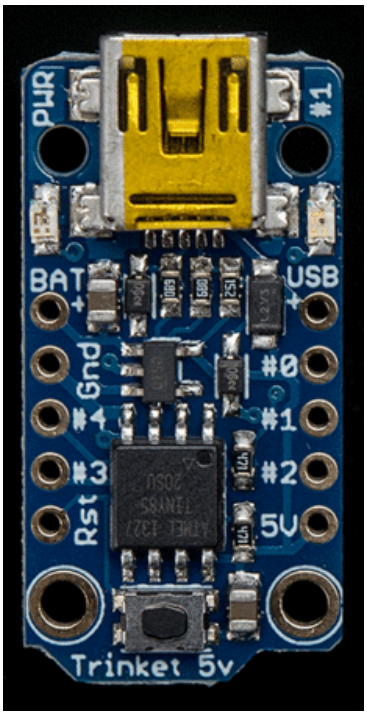

Abbildung 5: Oberseite Adafruit Trinket (zeigt 5V-Version des Trinket, zum Einsatz muss die 3,3 V-Version kommen)

D.h. das kurze Ende der Beinchen steht nach oben. Ergebnis: Adafruit Trinket mit Beinchen:

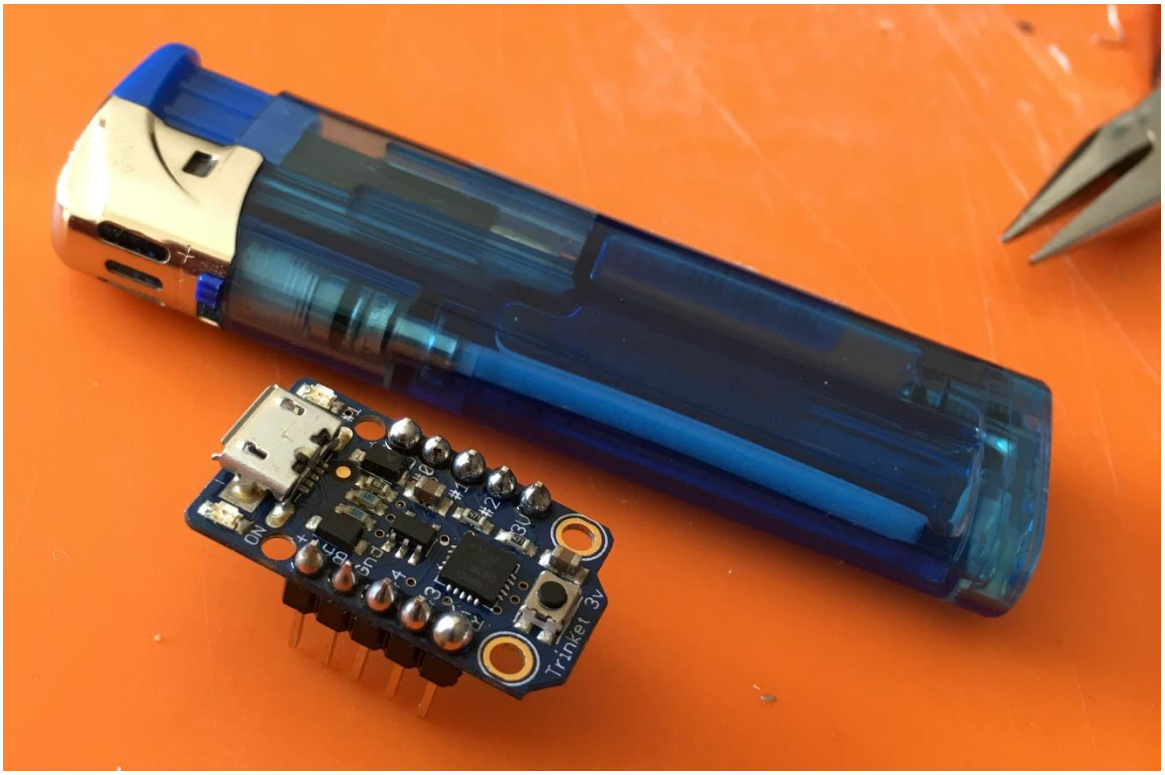

Abbildung 6: Adafruit Trinket mit Beinchen mit Größenvergleich

6.3.2 Vorbereiten der Europlatine

Entweder wir nehmen ein Reststück oder wir schneiden uns von einer Europlatine ein Stück mindestens im Format 35x30 (BxH) mm ab. Natürlich kann es auch größer sein.

6.3.3 Einlöten des Adafruit Trinket

Wir löten den Adafruit Trinket am besten an den PINS, die wir nicht benötigen, provisorisch ein. Hierzu können wir (von der Unterseite gesehen), die PINs oben links und unten rechts verwenden (roter Kreis):

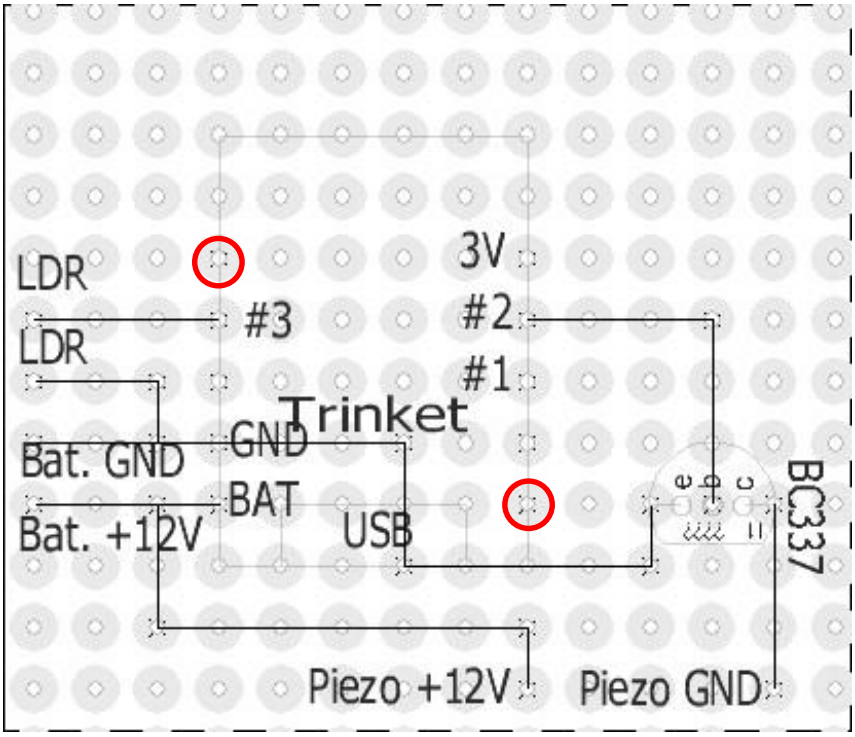

Abbildung 7: PINS zur Fixierung des Adafruit Trinket

6.3.4 Verlöten der Leiterbahnen

Anschließend verlöten wir die Leiterbahnen. Hierbei arbeiten wir uns von links nach rechts. Die Leiterbahnen bleiben an den äußeren Enden (Piezosignalgeber, Batterie, Fotowiderstand) erst einmal unfixiert, weil wir hier zum Schluss die Kabelenden der jeweiligen externen Komponenten anlöten werden.

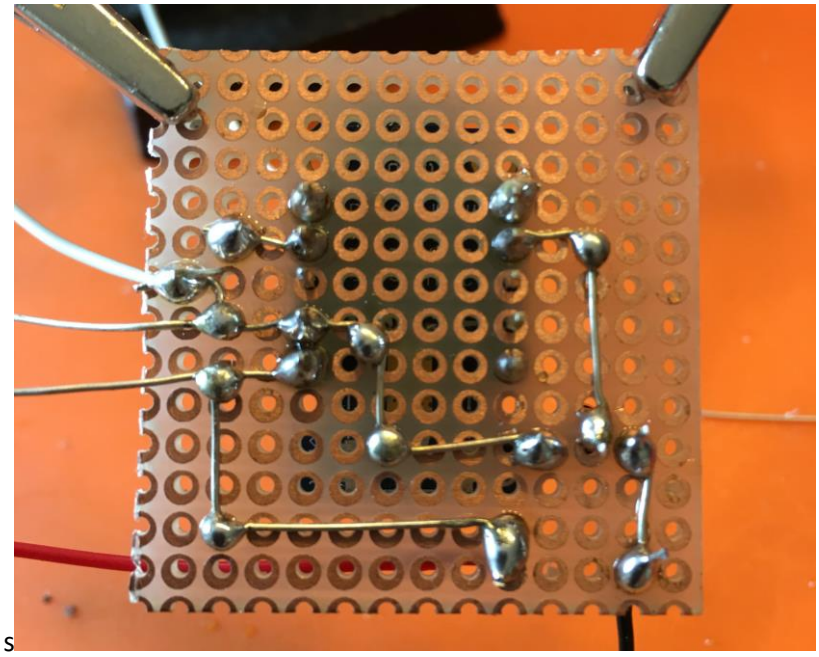

Abbildung 8: Europlatine mit gelöteten Leiterbahnen (es geht auch schöner)

6.3.5 Verlöten des BC337

Wir verlöten den npn-Transistor BC 3337 laut Plan.

6.3.6 Anlöten von zwei kurzen Kabeln an den Piezosignalgeber

Wir löten nun an die Anschlüsse des Piezosignalgebers zwei kurze Drähte an und schützen ggf. seine Enden am Gehäuse mit einem Schrumpfschlauch. Außerdem löten wir zwei Drähte mit der Platine an die Platine. Die Verbindung der vier Drähte passiert nach Einbau des Piezo-Signalgebers im Gehäuse.

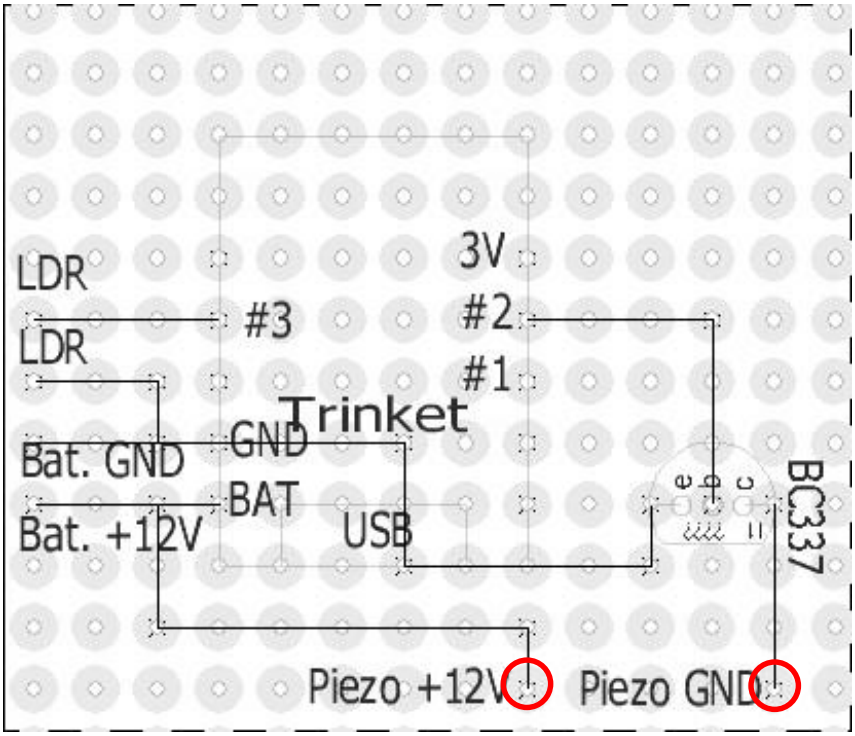

Abbildung 9: Anlöten Piezosignalgeber an Platine

6.3.7 Anlöten von zwei kurzen Kabeln an den Fotowiderstand

Wir löten nun an die Anschlüsse des Fotowiderstands zwei kurze Drähte an und schützen ggf. seine Enden am Gehäuse mit einem Schrumpfschlauch. Außerdem löten wir zwei Drähte mit der Platine an die Platine. Die Verbindung der vier Drähte passiert nach Einbau des Fotowiderstands im Gehäuse.

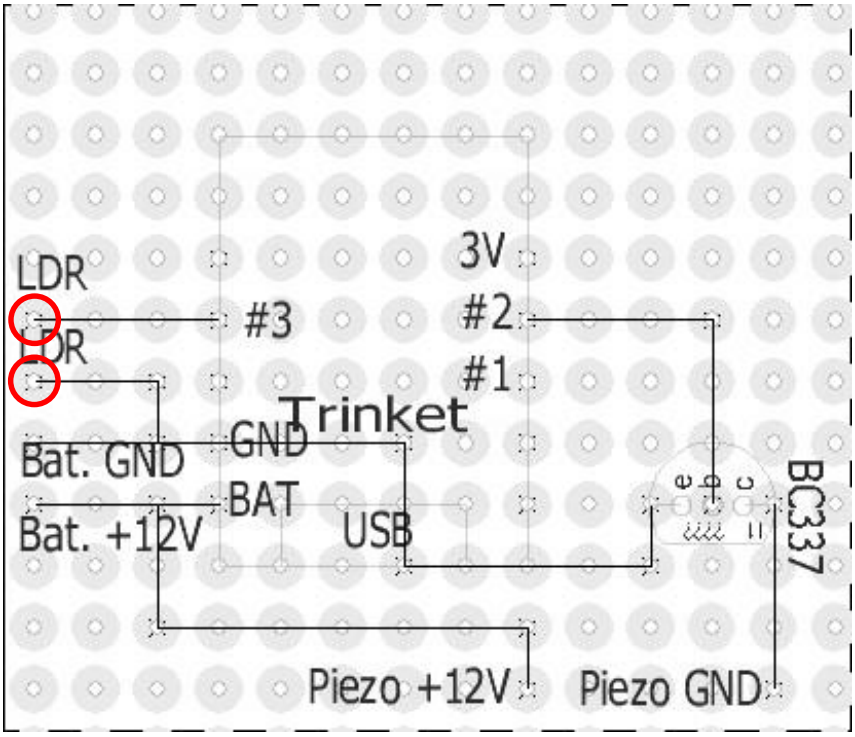

Abbildung 10: Anlöten LDR an Platine

6.3.8 Zusätzliche Modifikationen zur Laufzeitverlängerung

Grundlegend ist das Einsatz ja nicht „batteriekritisch“, da er nur über wenige Tage öder Nächte erfolgt und sich somit genug Möglichkeiten zur Batterieaufladung ergeben. Nichtsdestotrotz kann man den Energieverbrauch des Systems noch einmal deutlich reduzieren, in dem man eine oder zwei Modifikationen vornimmt, die allerdings einen feinen Lötkolben und Fingerspitzengefühl voraussetzen. Besondere Stromfresser sind nämlich die beiden auf der Platine des Adafruit Trinket vorhandenen LEDs.

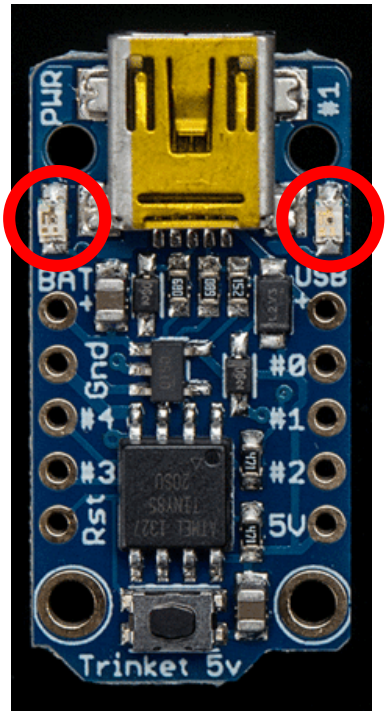

Abbildung 11: Adafruit Trinket LEDs

Beide LEDs verbrauchen im Betrieb einen Strom von ca. 1 mA pro Stunde. Gerade die linke, grüne Power-LED leuchtet permanent und zählt somit während der Ruhephasen zum Hauptstromverbraucher. Löten Sie diese aus, reduzieren Sie den Stromverbrauch um ca. ein Drittel. Das hat natürlich den Nachteil, dass sie beim Anschluss des Trinket nicht sehen, ob er „an“ ist.
Die rechte (rote) LED ist nicht so kritisch, das sie fast nicht verwendet wird und sie bei der Programmierung gute Dienste leistet, um zu erkennen, ob der Trinket im Programmiermodus ist. Aber es geht auch ohne und: Was nicht mehr auf der Platine ist, kann keinen Strom mehr verbrauchen.

6.4 Einbau in Gehäuse

Als erstes Testgehäuse wurde der in der Stückliste angegebene Verteilerkasten genutzt. Wir brauchen 4 Öffnungen:

- Eine mit ca. 32 mm Durchmesser für den Pirzosignalgeber
- Zwei in Drahtstärke für die Beine des Helligkeitssensors
- Eines für die Kabelzuführung zur Batterie

6.4.1 Bohrung für Piezosignalgeber

Das Loch für den Piezosignalgeber bohrt man am besten mit einem Forstnerbohrer in 32 mm Breite. Für den Prototypen hatte ich nur einen in 30 mm im Haus. Das ging auch, aber nicht ganz so schön.
Man steckt den Piezosignalgeber durch und kontert ihn mit der beiliegenden Mutter. Idealerweise dichtet man den Übergang zwischen Signalgeber und Gehäuse noch mit etwas Silikon ab. Dann verblöten wir die beiden Kabel mit den auf der Platine bereits angebrachten Kabel oder verbinden sie direkt mit der Platine (Plus und Minus beachten!).

6.4.2 Einbau Fotowiderstand

Für den Fotowiderstand bohrt man mit einem kleinen Nagel o.ä. zwei Löcher im entsprechenden Abstand durch die Membran am Boden. Dann verblöten wir die beiden Kabel mit den bereits auf der Platine angelötetenen Kabel oder verbinden Sie direkt mit der Platine.

6.4.3 Einbau Platine

Wenn Sie die Platine fest in das Gehäuse einbauen, achten Sie darauf, dass der USB-Anschluss so zugänglich bleibt, dass Sie jederzeit auch in Zukunft ein USB-Kabel anschließen können. So können Sie in Zukunft bei neuen Versionen der Software diese problemlos aufspielen, wie in den oberen Abschnitten beschrieben.

6.4.4 Bilder vom ersten Prototypen

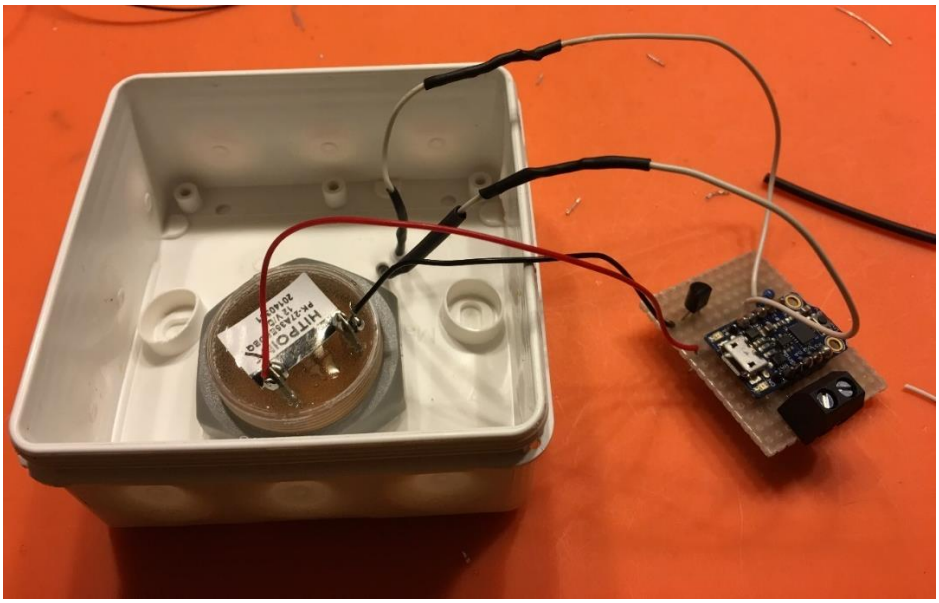

Abbildung 12: Erster Prototyp offen

Abbildung 13: Erster Prototyp Unterseite

6.5 Software aufspielen

Das Aufspielen der Software ist ein wenig knifflig, da es an einer Stelle zeitkritisch ist. Die folgenden Schritte sind erforderlich:

1. Herunterladen der Treiber für den Adafruit Trinket und Installieren der Treiber.
2. Adafruit Trinket anschließen.
3. Herunterladen der Software zur Programmierung.
4. Software „auspacken“.
5. Im Explorer in den Ordner wechseln.
6. Die Programmiersoftware starten und den Anweisungen dort folgen.

Die Schritte im Einzelnen:

6.5.1 Adafruit Trinket Treiber herunterladen und installieren

Laden Sie als erstes die Treiber für den Adafruit Trinket herunter https://github.com/adafruit/Adafruit_Windows_Drivers/releases/download/1.0.0.0/adafruit_drivers.exe und installieren Sie in. Folgen Sie am besten den Anweisungen des Herstellers: https://learn.adafruit.com/introducing-trinket/windows-setup

6.5.2 Anschluss des Adafruit Trinket

Schließen Sie nun den Adafruit Trinket an.

Hinweis:	Nutzen Sie, falls vorhanden, unbedingt einen USB 2.0-Port. Es soll mit USB 3.0 Ports schon Probleme gegeben haben. Falls Ihr Rechner nur USB 3.0 Ports haben sollte (diese erkennen Sie daran, dass sie innen blau gefärbt sind), nutzen Sie am besten einen (alten) USB-Hub. Nach dem Anschluss sollte die grüne Diode dauerhaft leuchten.

	Wenn Sie noch nichts verlötet haben, blinkt die rote Diode eine Zeitlang (ca. 10 Sekunden) schnell und anschließend langsam (ca. im 1-Sekunden-Takt). Wenn Sie den Piezosignalgeber verlötet haben, blinkt die rote Diode nicht, aber sie hören, wenn Sie alles richtig gemacht haben, schon erste Töne, eine Zeitlang (ca. 10 Sekunden) kurze, schnelle Töne und anschließend langsamer (ca. im 1-Sekunden-Takt).

6.5.3 Programmier-Software herunterladen

Die Software finden Sie unter http://trapcontrol.de/bambicontrol. Laden Sie sich das Softwarepaket herunter.
Die Software ist in ein Zip-Archiv verpackt. Dieses enthält die Firmware für BambiControl, eine Programmiersoftware („avrdude“) und eine „Batch-Datei“, die die Programmierung startet.

Hinweis:	Es kann sein, dass Ihr Virenscanner die Ausführung der Batch-Datei oder von „avrdude“ unterbindet. Bitte prüfen Sie die Dokumentation des Virenscanners, wie Sie die Datei zur Ausführung zulassen.

Wichtig:	Merken Sie sich den Ordner, wo sie die Software entpackt haben. Öffnen Sie jetzt einen Explorerfenster, dass Ihnen den Inhalt dieses Ordners anzeigt. Suchen Sie die Batch-Datei „burnBambiControl.cmd“ und starten Sie „burnBambiControl.cmd“ durch Doppelklick.

6.5.4 Programmierung

Nachdem die Software gestartet ist, sehen Sie zunächst das folgende Fenster:

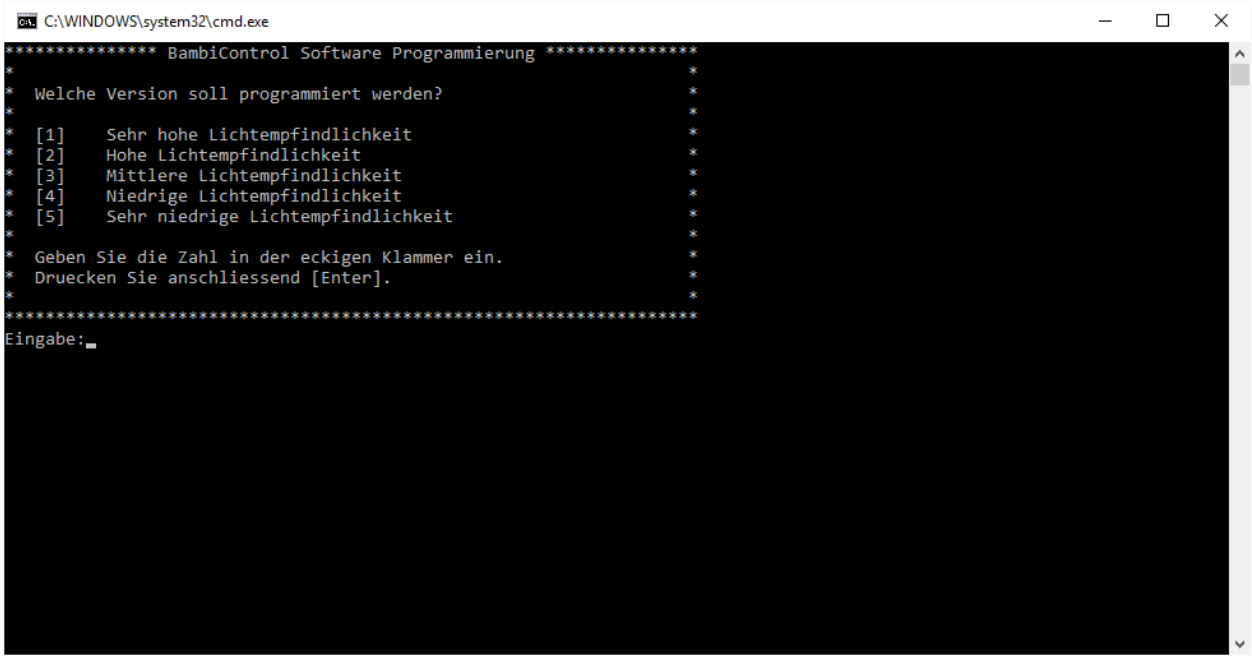

Abbildung 14: Startbildschirm der Programmiersoftware

Geben Sie nun die Ziffer Ihrer Wahl, die Sie zwischen den eckigen Klammern finden, ein und drücken Sie die „Return“- oder die „Enter“-Taste. Es erscheint das folgende Fenster:

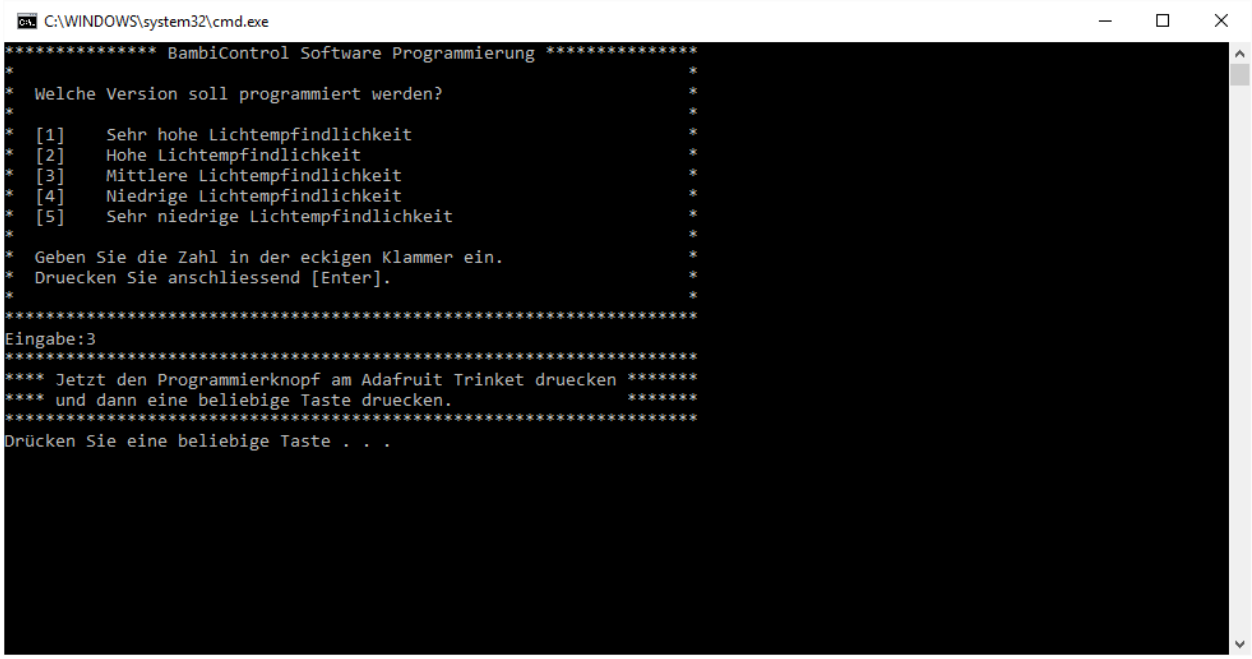

Abbildung 15: Programmiersoftware: Nächster Schritt

Um den Adafruit Trinket programmieren zu können, muss er **jetzt** in den Programmiermodus versetzt werden. Dies erreicht man durch kurzes Drücken des Programmierknopfes. Er befindet sich am anderen Ende der Platine, gegenüber des USB-Ports:

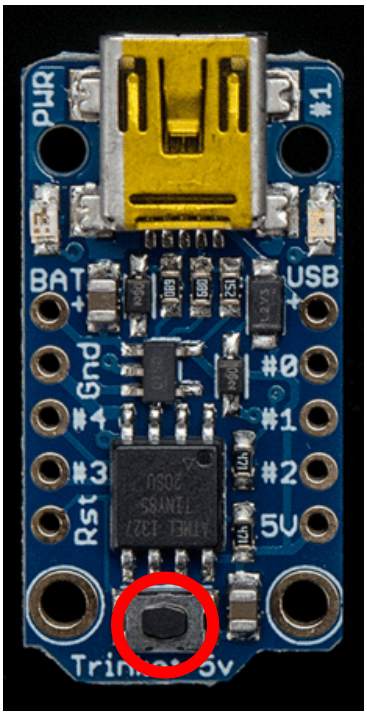

Abbildung 16: Adafruit Trinket Programmierknopf (roter Kreis)

Drücken Sie nun den Programmierknopf. Nachdem Sie den Knopf gedrückt haben, beginnt die rote LED für 10 Sekunden etwas schneller zu blinken. ***Nur in diesen 10 Sekunden*** befindet sich der Adafruit Trinket im Programmiermodus.

Drücken Sie nun eine beliebige Taste an Ihrem Computer und die Programmierung wird gestartet.

Hinweis:	Wenn Sie den Ton an ihrem Rechner eingeschaltet haben, hören Sie einen Ton, wenn der Adafruit Trinket von Ihrem Rechner erkannt wird, er also im Programmiermodus ist. Es ist der gleiche Ton, den Sie hören, wenn Sie irgendein Gerät (Maus, Tastatur, USB-Drucker, …) an ihren Rechner anschließen. Verlässt der Adafruit Trinket den Programmiermodus, hören Sie wieder einen Ton, nämlich denjenigen, wenn Sie ein Gerät von Ihrem Rechner entfernen (z.B. einen USB-Drucker ausschalten).

Die Programmierläuft nun durch und Sie sehen, hoffentlich, mehr oder weniger das folgende Fenster:

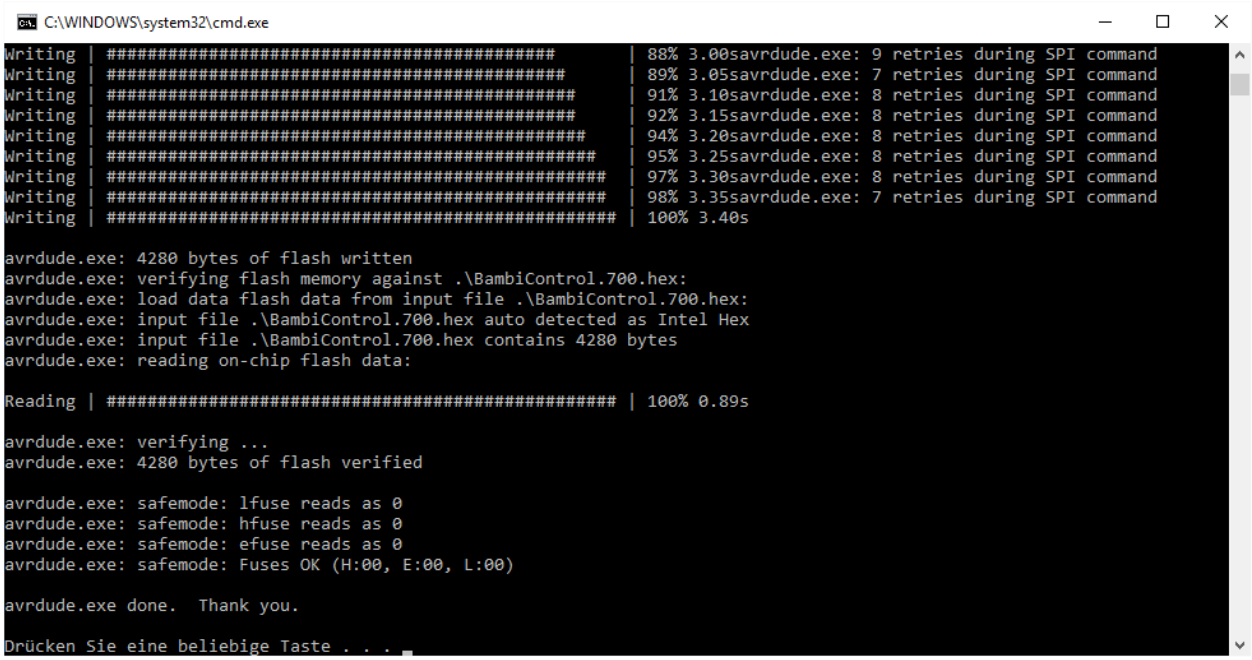

Abbildung 17: Ende der Programmierung

6.5.4.1 Fehlerhafte Programmierung

Eine fehlerhafte Programmierung erkennen Sie an der Ausgabe der Batchdatei.

```
C:\WINDOWS\system32\cmd.exe

avrdude.exe: Version 6.0.1, compiled on Apr 15 2015 at 19:59:58
             Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
             Copyright (c) 2007-2009 Joerg Wunsch

             System wide configuration file is ".\avrdude.conf"

             Using Port                    : usb
             Using Programmer              : usbtiny
avrdude.exe: Error: Could not find USBtiny device (0x1781/0xc9f)

avrdude.exe done.  Thank you.

Drücken Sie eine beliebige Taste . . .
```

Abbildung 18: Fehlerbeispiel 1: Adafruit Trinket nicht gefunden

Fehlerbehebung:

1. Treiber installieren (siehe oben).
2. Schneller sein. Zwischen dem Drücken des Programmierknopfes und dem Starten der Batchdatei dürfen maximal 10 Sekunden liegen.

```
C:\WINDOWS\system32\cmd.exe

avrdude.exe: 3276 bytes of flash written
avrdude.exe: verifying flash memory against .\KitzControl.cpp.hex:
avrdude.exe: load data flash data from input file .\KitzControl.cpp.hex:
avrdude.exe: input file .\KitzControl.cpp.hex auto detected as Intel Hex
avrdude.exe: input file .\KitzControl.cpp.hex contains 3276 bytes
avrdude.exe: reading on-chip flash data:

Reading |                                                  | 0% 0.00s
avrdude.exe: error: usbtiny_receive:  (expected 128, got -5)
Reading | #                                                | 1% 0.00s
avrdude.exe: error: usbtiny_receive:  (expected 4, got -5)
avr_read(): error reading address 0x0000
    read operation not supported for memory "flash"
avrdude.exe: failed to read all of flash memory, rc=-2


avrdude.exe: error: usbtiny_receive:  (expected 4, got -5)

avrdude.exe: error: usbtiny_receive:  (expected 4, got -5)

avrdude.exe: error: usbtiny_receive:  (expected 4, got -5)

avrdude.exe: error: usbtiny_receive:  (expected 4, got -5)
avrdude.exe: safemode: Sorry, reading back fuses was unreliable. I have given up and exited programming mode

avrdude.exe: error: usbtiny_transmit:

avrdude.exe done.  Thank you.

Drücken Sie eine beliebige Taste . . .
```

Abbildung 19: Fehlerbeispiel 2: USB 3.0 Port

Fehlerbehebung:

- Anderen, langsameren USB-Port verwenden

6.5.4.2 Erfolgreiche Programmierung

Wenn die Programmierung erfolgreich abläuft, sehen Sie eine ganze Reihe von kryptischen Ausgaben:

```
C:\WINDOWS\system32\cmd.exe
Writing | ##########################################          | 84% 2.74savrdude.exe: 9 retries during SPI command
Writing | ##########################################          | 86% 2.79savrdude.exe: 8 retries during SPI command
Writing | ##########################################          | 88% 2.84savrdude.exe: 8 retries during SPI command
Writing | ##########################################          | 90% 2.89savrdude.exe: 8 retries during SPI command
Writing | ##########################################          | 92% 2.94savrdude.exe: 8 retries during SPI command
Writing | ##########################################          | 94% 2.99savrdude.exe: 8 retries during SPI command
Writing | ##########################################          | 96% 3.04savrdude.exe: 8 retries during SPI command
Writing | ##########################################          | 98% 3.09savrdude.exe: 8 retries during SPI command
Writing | ##########################################          | 100% 3.14s

avrdude.exe: 3276 bytes of flash written
avrdude.exe: verifying flash memory against .\KitzControl.cpp.hex:
avrdude.exe: load data flash data from input file .\KitzControl.cpp.hex:
avrdude.exe: input file .\KitzControl.cpp.hex auto detected as Intel Hex
avrdude.exe: input file .\KitzControl.cpp.hex contains 3276 bytes
avrdude.exe: reading on-chip flash data:

Reading | ##########################################          | 100% 0.73s

avrdude.exe: verifying ...
avrdude.exe: 3276 bytes of flash verified

avrdude.exe: safemode: lfuse reads as 0
avrdude.exe: safemode: hfuse reads as 0
avrdude.exe: safemode: efuse reads as 0
avrdude.exe: safemode: Fuses OK (H:00, E:00, L:00)

avrdude.exe done.  Thank you.

Drücken Sie eine beliebige Taste . . .
```

Abbildung 20: Erfolgreiche Programmierung

Hinweis:	Achten Sie auf Ausgabe (im hier eingezeichneten roten Kasten) von
	
	`avrdude.exe: verifying ...`
	`avrdude.exe: 3276 bytes of flash verified`
	
	Die Zahl „3276“ kann auch einen anderen Wert aufweisen. Es kommt auf die Stichworte „flash verified“ an. Wenn das erscheint, haben Sie den Adafruit Trinket erfolgreich programmiert.
	
	Falls Sie BambiControl schon fertig zusammengelötet haben, piepst außerdem der Signalgeber deutlich hörbar unterschiedlich zwei bis dreimal zur Bestätigung.

7 Betrieb und Einsatz

7.1 Inbetriebnahme

BambiControl wird einfach dadurch in Betrieb genommen, in dem man es an der Batterie (12 V) anschließt. Hierbei auf die richtige Polung („+“ an „+“ und „–“ an „–“) achten.

7.2 Arbeitsweise

Nach dem Einschalten von BambiControl wartet BambiControl zunächst einige Sekunden, damit Sie sich entfernen können, um Ihr Gehör zu schonen.

WARNUNG: Der vorgeschlagene Piezosignalgeber ist mit 95 dB sehr laut. Wenn Sie das Gerät aufstellen, sollten Sie die Nutzung eines Gehörschutzes in Erwägung ziehen!

Danach gibt BambiControl unterschiedliche Signale ab, um einen möglichen Fehler oder seine Betriebsbereitschaft zu signalisieren.

Fehlersignal:	Kurz – Kurz – Kurz – Lang – Lang – Lang – Kurz – Kurz – Kurz (das „SOS“-Signal): Hiermit signalisiert BambiControl, das irgendetwas nicht stimmt, also das beispielsweise der vom Fotowiderstand zurückgemeldete Wert nicht stimmig ist, weil der Fotowiderstand (vermeintlich) falsch angeschlossen oder defekt ist. Aber: Wenn Sie BambiControl in der Dunkelheit oder Dämmerung anschalten, liest BambiControl natürlich auch einen eigentlich „unstimmigen“ Wert aus und meint, der Fotowiderstand arbeite nicht richtig. In diesem Fall können Sie das Fehlersignal vorerst ignorieren, sollten am nächsten Tag bei Helligkeit jedoch BambiControl mal kurz die Batterie klauen und wieder anschließen, um die Funktionsfähigkeit des Fotowiderstands sicherzustellen.

Hinweis:	Wenn ein Fehlersignal auftritt, arbeitet BambiControl trotzdem, jedoch unter Umständen unabhängig von der Helligkeit, beispielsweise wenn der Fotowiderstand falsch angeschlossen oder defekt ist. D.h. in diesem Fall werden nicht nur „nachts“ Signale abgegeben, sondern den gesamten Tag. Wenn die Fehlermeldung nur damit zusammenhängt, dass Sie BambiControl bei Dunkelheit aktiviert haben, arbeitet BambiControl dann wahrscheinlich trotzdem wie geplant, d.h. morgens, nach Erreichen einer bestimmten Helligkeit, endet die Signalisierung.

„Bereit“-Signal:	Ertönen 10 kurze Signale (0,25 Sekunden) mit einer Pause von 0,75 Sekunden: BambiControl ist bereit.

Anschließend wartet BambiControl ab, bis eine gewisse Dunkelheit eingetreten ist. Sofern ein Schwellenwert erreicht ist (den Sie selbst über die unterschiedlichen Softwarevarianten wählen können)

beginnt BambiControl seine Signale abzusetzen, unterbrochen von einer Pause. Es werden also Signalphasen und Pausenphasen unterschieden und diese wechseln sich ab.

7.2.1 Signalphase

Eine Signalphase dauert zwischen 30 und 180 Sekunden. Die Dauer wird vor jeder Phase per Zufall ermittelt. Diese Signalphase wird dann aufgeteilt, um keinen einfachen Dauerton und/oder Dauerlichtsignal zu erzeugen, sondern möglichst unterschiedliche Signale. Es ertönen bis zu 60 Signale unterschiedlicher Länge mit unterschiedlichen Pausen.

7.2.2 Ruhephase

Jeder Signalphase schließt sich eine Ruhephase an. Die Dauer dieser Ruhephase ist ebenfalls zufallsgesteuert. Sie liegt zwischen 900 und 1500 Sekunden (also 15 und 25 Minuten).
Nach jeder Ruhephase kommt dann wieder eine Signalphase, bis der Schwellenwert der Helligkeit überschritten wird, BambiControl also der Auffassung ist, dass es nun Tag ist.

7.3 Wirkungsbereich

Der vorgeschlagene Piezosummer hat laut Hersteller einen Schallpegel von 95 dB in einem Abstand von 30 cm. Dies ergibt die folgenden theoretischen Werte der Lautstärke bei unterschiedlichen Entfernungen vom Aufstellort:

Entfernung vom Aufstellort	Schallpegel (ca.)	Vergleichbar mit
30 cm	95 dB	Lastwagen (90 dB) Ghettoblaster (100 dB)
10 m	78 dB	Presslufthammer (80 dB)
25 m	72 dB	Fernseher, Schreien, Rasenmäher (70 dB)
50 m	66 dB	Kantinenlärm (65 dB)
75 m	64 dB	Kantinenlärm (65 dB)
100 m	62 dB	Nähmaschine, Gruppengespräch (60 dB)
150 m	59 dB	Nähmaschine, Gruppengespräch (60 dB)
200 m	57 dB	Normales Gespräch (55 dB)

Dies sind natürlich theoretische Werte bei Übertragung durch Luft ohne Hindernisse (Regen, Bewuchs, ---). Nichtsdestotrotz kann man von einer Wirksamkeit von ca. 100 bis 150 Metern ausgehen.