

Benutzerhandbuch

Dokumentation

Aufbauanleitung

---

# EL FUERADORO

Funkgesteuertes Zündsystem für Feuerwerke

---

Felix Pflaum ([f.pflaum@gmail.com](mailto:f.pflaum@gmail.com))  
31.08.2020, 20:20



# Wichtige Hinweise

Vor Inbetriebnahme sollte in jedem Fall diese Anleitung gelesen und verstanden werden! Speziell die folgenden, weiter hinten in der Anleitung detaillierter ausgeführten Hinweise müssen beachtet werden, um Personenschäden oder Beschädigungen an den Geräten zu vermeiden:

- 1. Bei den Zündboxen der ersten Generation liegt das Metallgehäuse des SMA-Antennenanschlusses an der Oberseite der Zündbox auf Massepotential, die roten Anschlussklemmen nach dem Einschalten und unter Umständen auch noch Minuten nach dem Ausschalten bei 22,5 V. Ein Kurzschluss zwischen diesen beiden Punkten ist daher unbedingt zu vermeiden, um Schäden an der Schaltung und ungewollte Zündungen zu verhindern. Daher, sofern der Antennenanschluss nicht isoliert ist, beim Anschließen die Anzünder immer zuerst mit der schwarzen Klemme, danach mit der roten verbinden, beim Abklemmen zuerst das Kabel an der roten, dann das an der schwarzen Klemme lösen. Weiterhin sollte man die Zündbox erst in ihrer endgültigen Position einschalten, so dass ein unbeabsichtigtes Herausziehen der Kabel tunlichst ausgeschlossen ist. Sicherheitshalber kann auch der SMA-Stecker nach dem Anschrauben mit Klebeband isoliert werden. Bei den Zündboxen ab der zweiten Generation sind schaltungstechnische Maßnahmen getroffen, um den Strom im Kurzschlussfall zu begrenzen und ein Auslösen der Anzünder bei Verbindung zwischen roter Klemme und Antennenanschluss im nicht-scharfgeschalteten Zustand auszuschließen.**
- 2. Devices niemals ohne angeschlossene Antenne einschalten, um Rückreflexionen an offenen Steckern/Buchsen zu vermeiden, welche die Sendeendstufe zerstören könnten.**
- 3. Keine Softwareupdates an Zündboxen durchführen, solange Anzünder angeschlossen sind, da das Verhalten nach einem fehlgeschlagenen Update unvorhersehbar ist.**

Trotz aller Sorgfalt bei der Entwicklung von Hard- und Software sowie ausgiebigen Tests seitens der Entwickler kann keinerlei Garantie für Sicherheit und Funktionalität von *El Fueradoro* sowie keine Haftung für Sach- und Personenschäden, welche sich direkt oder indirekt aus dem Einsatz von *El Fueradoro* ergeben, übernommen werden. Es werden mit *El Fueradoro* dem Anwender lediglich Schaltpläne, Layouts und Programmcode sowie Hinweise zu Aufbau und Verwendung einer elektronischen Funkzündanlage, welche er frei weiter verbreiten und – selbstverständlich auf eigene Gefahr – verändern darf, zur Verfügung gestellt. *El Fueradoro* ist und bleibt ein Hobby-Bastelprojekt, kein geprüftes oder in irgendeiner Form zertifiziertes Produkt!

Der Umgang mit Feuerwerkskörpern und explosivem Material unterliegt in Deutschland und der

EU harmonisierten wie auch nationalen gesetzlichen Restriktionen, deren Einhaltung im Verantwortungsbereich des Anwenders liegt. *El Fueradoro* darf nicht von Personen unter 18 Jahren bedient werden.

# Inhaltsverzeichnis

<b>Wichtige Hinweise</b>	<b>3</b>
<b>Inhaltsverzeichnis</b>	<b>5</b>
<b>I Benutzerhandbuch</b>	<b>9</b>
<b>1 Das System</b>	<b>11</b>
1.1 Aufbau . . . . .	11
1.2 Transmitter und Zündboxen . . . . .	14
1.3 Energieversorgung . . . . .	15
1.4 Die Status-LEDs . . . . .	15
1.5 Das LCD des Transmitters . . . . .	16
1.6 Schalter an den Zündboxen . . . . .	17
<b>2 Vorbereitung des PCs</b>	<b>19</b>
2.1 Installation eines USB-RS232-Adapters . . . . .	19
2.2 Einrichtung des Terminalprogramms . . . . .	21
2.3 Einrichtung von <i>Pyro Ignition Control</i> . . . . .	22
<b>3 Kommunikation zwischen PC und Devices</b>	<b>25</b>
3.1 Puttytel . . . . .	25
3.1.1 Befehlsübersicht . . . . .	25
3.1.2 Konfiguration . . . . .	26
3.1.3 Systemübersicht . . . . .	29
3.1.4 Manuelles Senden . . . . .	29
3.1.5 Funkmodul-Zugriff . . . . .	30
3.1.6 AES-Verschlüsselung . . . . .	33
3.2 GUI . . . . .	34
3.3 Firmwareupdate . . . . .	35
3.3.1 Herunterladen . . . . .	36
3.3.2 Aktualisierung . . . . .	36
<b>4 raspEasyFire</b>	<b>39</b>
4.1 Erster Start . . . . .	39
4.2 Konfiguration und Zugriff . . . . .	41
4.3 Anpassung und Umbenennung von AES-Schlüsseln . . . . .	41

4.4	Passwort ändern . . . . .	43
4.5	Zündpläne und Musikdateien übertragen . . . . .	43
4.5.1	USB-Stick . . . . .	43
4.5.2	SFTP . . . . .	43
4.6	Die Benutzeroberfläche . . . . .	44
4.6.1	Zündplanauswahl . . . . .	45
4.6.2	Slavesuche . . . . .	46
4.6.3	Weitere Optionen . . . . .	46
4.6.4	Dateien löschen . . . . .	49
4.6.5	Soundcheck . . . . .	50
4.6.6	Widerstandsmessung . . . . .	50
4.6.7	Schlüsselwörter in Zündplänen . . . . .	51
4.6.8	Showstart . . . . .	51
<b>5</b>	<b>Show</b>	<b>55</b>
5.1	Vorbereitung . . . . .	55
5.1.1	Beidseitiges Zünden . . . . .	55
5.1.2	Zündkreisauslegung . . . . .	55
5.1.3	Funkreichweite . . . . .	58
5.1.4	Erstellung und Überprüfung des Zündplans . . . . .	59
5.2	Durchführung . . . . .	59
<b>II</b>	<b>Dokumentation</b>	<b>61</b>
<b>6</b>	<b>Schaltpläne &amp; Layouts</b>	<b>63</b>
<b>7</b>	<b>Datenblätter</b>	<b>77</b>
<b>8</b>	<b>Pinbelegung</b>	<b>79</b>
8.1	Mikrocontroller . . . . .	79
8.2	Raspberry-Pi-Aufsteckplatine . . . . .	80
<b>9</b>	<b>Software</b>	<b>83</b>
<b>III</b>	<b>Aufbauanleitung</b>	<b>85</b>
<b>10</b>	<b>Materiallisten</b>	<b>87</b>
<b>11</b>	<b>Platinenherstellung</b>	<b>99</b>
<b>12</b>	<b>Aufbau</b>	<b>103</b>
12.1	PC-Transmitter und Zündbox . . . . .	103
12.1.1	Kabel . . . . .	103
12.1.2	Platinen . . . . .	104
12.2	Raspberry-Pi-Aufsteckplatine . . . . .	106

12.3 Funkmodul-Adapter . . . . .	107
12.4 Peripherie . . . . .	108
12.4.1 Transmitter . . . . .	108
12.4.2 Zündbox . . . . .	110
12.4.3 Koffer . . . . .	114
12.4.4 Gehäuse des Raspberry-Pi-Transmitters . . . . .	118
<b>13 Aufspielen des Bootloaders</b>	<b>119</b>
13.1 Verwendung des AVRISP mkII . . . . .	119
13.2 Verwendung eines anderen Programmieradapters . . . . .	120
<b>14 Micro-SD-Karte für raspEasyFire vorbereiten</b>	<b>123</b>
14.1 Image herunterladen und übertragen . . . . .	123
14.2 WLAN-Daten einstellen . . . . .	124
<b>15 Tipps und Tricks</b>	<b>127</b>
15.1 5V-LCD an 3,3V . . . . .	127
15.2 Antennenbau . . . . .	128
15.3 Kompilieren der Firmware . . . . .	131
15.4 Umrechnung zwischen Watt und dBm . . . . .	131
<b>Abbildungsverzeichnis</b>	<b>133</b>
<b>Tabellenverzeichnis</b>	<b>135</b>
<b>Danksagung</b>	<b>137</b>



# **Teil I**

# **Benutzerhandbuch**



# 1 Das System

*El Fueradoro* ist eine Eigenentwicklung zur automatisierten Zündung von Feuerwerkschoreographien per Computer und Funk, der Name *El Fueradoro* leitet sich als Kofferwort aus den spanischen Wörtern *fuego*, *radio* und *oro* her und bedeutet frei übersetzt daher so etwas wie „Goldenes Funkfeuer“. Sie wurde zur Verwendung mit der frei verfügbaren Software *Pyro Ignition Control* von Yannic Wilkening (Version 1.4.5 und später) geplant.

Das ursprüngliche System bestand aus einer mit einem PC verbundenen Transmitterbox, welche die von *Pyro Ignition Control* generierten Zündbefehle per Funk an die Zündboxen weitergibt, und einer oder mehreren Zündboxen mit je 16 oder 32 Kanälen zur elektrischen Zündung. Dieser klassische Aufbau des Systems ist schematisch in Abbildung 1.1 gezeigt. Die reine Transmitterbox wird nicht zwingend benötigt; es kann alternativ auch eine Zündbox per Kabel mit dem PC verbunden werden und neben ihrer originären Aufgabe auch die Transmitteraufgaben bewältigen.

Mittlerweile hat sich als Standalone-Lösung, also als Ersatz für Transmitterbox und PC, der Raspberry-Pi-Transmitter *raspEasyFire* etabliert, welcher einen deutlich größeren Funktionsumfang bietet als die ursprüngliche Variante.

Um möglichst große Flexibilität bei der Gestaltung eines Feuerwerks mit *El Fueradoro* zu gewährleisten, was z. B. die Anzahl der verwendeten Zündboxen oder auch die Tatsache angeht, dass man identische Effekte gerne zeitgleich an zwei weiter voneinander entfernten Orten zünden möchte, können den Boxen per Software zwei Kennnummern (Unique-ID und Slave-ID) zugewiesen werden. Genaueres dazu findet sich im Abschnitt 3.

Transmitterbox und Zündbox sind in Abbildung 1.2a bzw. 1.2b mit ihren wesentlichen Bestandteilen gezeigt. Sie werden im weiteren Text unter dem Begriff „Devices“ zusammengefasst, falls sich Aussagen auf beide Teile beziehen, und in den folgenden Kapiteln näher beschrieben.

Um den Raspberry-Pi-basierten Transmitter *raspEasyFire* geht es in Abschnitt 4 ab Seite 39.

## 1.1 Aufbau

Detailliertere Infos zum Aufbau erhalten interessierte Leser in den nächsten Teilen dieses Dokuments ab Seite 63. An dieser Stelle soll nur kurz auf die Funktionalität eingegangen werden. Herzstück aller

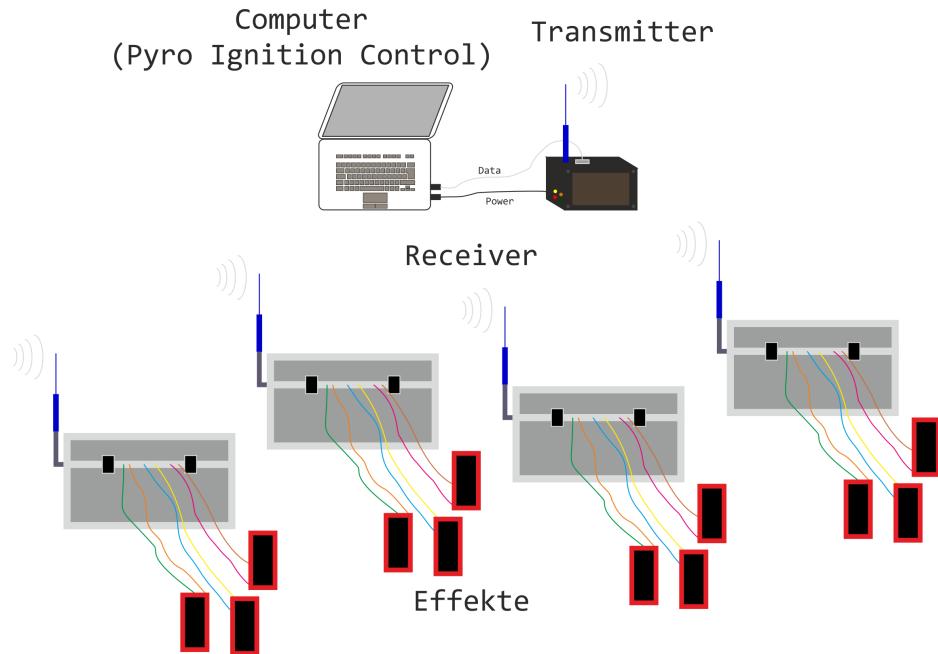


Abbildung 1.1: Systemübersicht *El Fueradoro*

Devices ist der 8-bit-Mikrocontroller **ATmega328P** von Atmel, welcher die verschiedenen Peripheriebausteine kontrolliert. In jedem Device befinden sich ein Funkmodul **RFM69CW** oder **RFM69HCW** von HopeRF, welches die drahtlose Kommunikation zwischen den einzelnen Devices übernimmt, ein RS232-Treiberbaustein **MAX202** mit an die Gehäuseaußenseite geführter Sub-D-Buchse bzw. ab der dritten Generation ein USB-UART-Transceiver **MCP2221** mit USB-Buchse zur Kommunikation zwischen Mikrocontroller und PC sowie vier Status-LEDs.

Die Transmitterbox ist mit einem LC-Display mit 20 Spalten und 4 Zeilen ausgestattet.

In den Zündboxen befinden sich zwei vom Mikrocontroller gesteuerte kaskadierte Schieberegister **74HC595**, welche mit ihren insgesamt 16 bzw. 32 Ausgängen 16 bzw. 32 Feldeffekttransistoren vom Typ **IRF3708** für die Zündung ansteuern, außerdem ist auf der Platine mit dem **MC33063** bzw. einem fertigen Modul mit dem **XL6009** ein Hochsetzsteller zur Erzeugung einer höheren Zündspannung integriert sowie ein Schlüsselschalter zum Scharfschalten der Boxen. Bei den ersten beiden Generationen liegt parallel zur Drain-Source-Strecke jedes Feldeffekttransistors eine LED mit Vorwiderstand, um zu signalisieren, an welchen Kanälen Anzünder angeschlossen sind. Ab der dritten Generation werden diese Kanal-LEDs über LED-Treiber **DM13A** angesteuert, um genauere Informationen über den Kanalzustand und weitere Funktionen darstellen zu können.



Abbildung 1.2: Die Devices von *El Fueradore*

## 1.2 Transmitter und Zündboxen

Die in Abbildung 1.2a gezeigte Transmitterbox dient dazu, die Zündbefehle des PCs per Funk an die über das Gelände verteilt stehenden Zündboxen weiterzuleiten sowie das komplette Funksystem zu überwachen. Hierzu ist im Inneren der Transmitterbox sowie der Zündboxen ein Funkmodul für das – im Rahmen der Vorschriften der jeweiligen nationalen Aufsichtsbehörde, in Deutschland der Bundesnetzagentur – frei nutzbare Frequenzband um 868 MHz verbaut. Die Datenübertragung vom PC geschieht über eine serielle Schnittstelle.

Die Zündboxen, in Abbildung 1.2b dargestellt, empfangen die Anweisungen und zünden die einzelnen Kanäle. Die hier gezeigte Variante der Zündbox enthält 16 einzeln ansteuerbare Zündkanäle. Jeweils eine rote und eine schwarze Klemme bilden zusammen einen Zündkanal, die Nummerierung beginnt links unten mit Kanal 1 und endet rechts oben mit Kanal 16. Alle roten Klemmen sind nach dem Einschalten unmittelbar mit einer Gleichspannung von 22,5 V verbunden, die schwarzen Klemmen mit dem Drain-Anschluss eines von 16 n-Kanal-MOSFETs.

Welches der beiden Kabel an welcher Klemme des Zündkanals angeschlossen wird, spielt keine Rolle, da es sich bei den Anzündern um passive elektrische Bauelemente handelt, die in beiden Richtungen gleichermaßen von Strom durchflossen werden können. Die Verdrahtung sollte aber stets im ausgeschalteten Zustand der Boxen erfolgen und aus Sicherheitsgründen immer in folgender Reihenfolge:

- Beim Anklemmen der Kabel zuerst ein Kabel an der schwarzen Klemme befestigen, anschließend das andere an der roten!
- Beim Lösen der Kabel zuerst das Kabel von der roten Klemme lösen, anschließend das von der schwarzen!

Dies ist der Tatsache geschuldet, dass bei der ersten Generation der Zündboxen, wenn eine Seite des Zündkanals fest mit der Zündspannung, d. h. einer roten Klemme verbunden ist, ein frei baumelndes zweites Kabel durch Berührung eines Bauteils auf Massepotential, z. B. der Antennenbuchse oder der Überwurfmutter des gewinkelten SMA-Steckers, einen Kurzschluss und somit eine Zündung auslösen würde – selbst dann, wenn die Box nicht scharf geschaltet ist! Ab der zweiten Generation der Zündboxen ist dieses Manko behoben.

Bei den Zündboxen der ersten und zweiten Generation ist parallel zur üblicherweise sperrenden Drain-Source-Strecke jedes MOSFETs eine grüne LED mit Vorwiderstand geschaltet, welcher den Strom durch den LED-Zweig auf unter 5 mA begrenzt, so dass ein Zünden über die LED ausgeschlossen ist. Sobald die zugehörige schwarze Klemme mit einer roten Klemme – die Zuordnung kann beliebig erfolgen, obwohl zwecks Übersichtlichkeit natürlich ratsam ist, die nebeneinander liegenden Klemmen zu nutzen – verbunden ist, also ein geschlossener Strompfad von 22,5 V zur Schaltungsmasse besteht, wird dies durch Leuchten der zur schwarzen Klemme gehörigen grünen LED signalisiert. Ein Leuchten der grünen LED ist dabei lediglich ein Indikator, dass „etwas“ am jeweiligen Kanal angeschlossen ist, eine Aussage, ob der Widerstand des angeschlossenen Anzündernetzwerks gering genug ist, um eine

Farbe	Funktion	Startup-Faktor
orange	Funkmodul empfängt	4
grün	Funkmodul sendet	2
gelb	Daten kommen über serielle Schnittstelle an	1
rot	Device ist scharf geschaltet	8

Tabelle 1.1: Farben und Funktionen der Status-LEDs

Zündung auszulösen, wird durch die LED ausdrücklich nicht getroffen.

Anders verhält sich dies bei den Zündboxen der dritten Generation: Hier werden die Widerstände der einzelnen Kanäle periodisch bestimmt und die zugehörige grüne LED nur dann aktiviert, wenn der Kanalwiderstand ausreichend gering ist.

## 1.3 Energieversorgung

Die in unmittelbarer Nähe des zu steuernden PCs platzierte Transmitterbox bezieht ihre nötige elektrische Energie aus dem USB-Port eines PCs. Das fest mit der Transmitterbox verbundene USB-Kabel dient ausschließlich diesem Zweck. Sie verfügt über keinen Ein/Aus-Schalter, sondern ist eingeschaltet, solange sie mit dem USB-Port verbunden und der PC eingeschaltet ist. Da der USB-Port „angezapft“ wird, ohne in der sonst üblichen Weise mit dem Controller zu kommunizieren, empfiehlt sich, die USB-Verbindung erst nach komplett abgeschlossenem Bootvorgang herzustellen und vor dem Herunterfahren des Rechners wieder zu trennen.

Aufgrund der anzunehmenden Platzierung der Zündboxen im freien Feld, abseits von Steckdosen und anderen Energiequellen werden sie über eine Batterie versorgt. **Um ordnungsgemäße Funktionalität zu garantieren und Schäden an der Schaltung zu vermeiden, muss die Batteriespannung zwischen 8 V und 15 V liegen!** Empfohlen wird die Verwendung eines Blei-Vlies-Akkus mit Nennspannung 12 V, wie er auch in Abbildung 1.2b über der Zündbox zu erkennen ist.

## 1.4 Die Status-LEDs

Alle Devices verfügen über vier Status-LEDs. Bei den Transmitterboxen liegen sie direkt neben dem Kabel für die Energieversorgung auf der Seite, bei den Zündboxen auf der Oberseite. Diese sind mit ihrer Bedeutung in Tabelle 1.1 aufgeführt und leuchten, wenn das jeweilige Device die mit der LED verknüpfte Tätigkeit ausführt.

Während des Bootvorgangs wird durch die Status-LEDs dem Benutzer die aktuell eingestellte Slave-ID der Box in Binärdarstellung visualisiert. Hierzu gehen zunächst nacheinander schnell die gelbe, grüne, orange und rote LED an, ehe zwei Sekunden lang eine Kombination der Status-LEDs aufleuchtet.

Anschließend gehen nacheinander schnell die rote, orange, grüne und gelbe LED an, bevor erneut zwei Sekunden lang eine weitere Kombination der Status-LEDs leuchtet. Der Zahlenwert, welcher durch eine Status-LED repräsentiert wird, ist in Tabelle 1.1 aufgeführt. Die eingestellte Slave-ID entspricht der Summe aus dem Wert der ersten Kombination multipliziert mit 16 und der zweiten Kombination.

Beispiel: Leuchtet nach dem ersten Lauflicht zunächst zwei Sekunden lang nur die gelbe LED und nach der zweiten Lauflichtsequenz leuchten zwei Sekunden die orange und grüne LED zusammen, ergibt sich die Slave-ID zu  $16 \cdot 1 + 4 + 2 = 22$ .

Ab der dritten Generation werden während dieser Sequenz auch alle Kanal-LEDs eingeschaltet, um defekte LEDs erkennen zu können.

Das nach kurzer Pause folgende Blinken der grünen LED signalisiert dann die Statusmeldung per Funk an allen anderen Devices.

Bei einem Zündvorgang leuchten an der Zündbox für die Dauer der Zündung (11 ms) alle vier LEDs, bei Zündboxen ab der dritten Generation zusätzlich auch die zum gezündeten Kanal zugehörige Kanal-LED.

## 1.5 Das LCD des Transmitters

Aktuelle Statusanzeigen werden beim Transmitter auf dem LC-Display ausgegeben. Ein Beispiel zeigt Abbildung 1.3. In Zeile 1 wird hinter der Abkürzung „Tx“ (Transmitted) der letzte gesendete Befehl angezeigt, in Zeile 2 – im Bild nicht zu sehen – ggf. hinter der Abkürzung „Rx“ (Received) die letzte empfangene Rückmeldung (angeforderte Parameter). Die am Ende der zweiten Zeile erscheinende negative Zahl steht für den RSSI-Wert beim Empfang in dBm.

In den Zeilen 3 und 4 werden die letzten sechs gesendeten Kommandos aufgelistet. Zwei zweistellige Zahlen getrennt durch ein Flammensymbol stehen dabei für einen Zündbefehl. Die erste Zahl gibt die Slave-ID, die zweite den zu zündenden Kanal an. Für den Fall, dass eine Aufforderung zur Identifikation gesendet wurde, erscheint „IDENT“, für eine Aufforderung zur Temperaturmessung „TEMP“. Das „x“ steht immer vor dem bis dato letzten Befehl, springt also mit jedem neuen Befehl eine Stelle weiter.

Alle Zeilen werden nach einer bestimmten Zeit automatisch gelöscht.

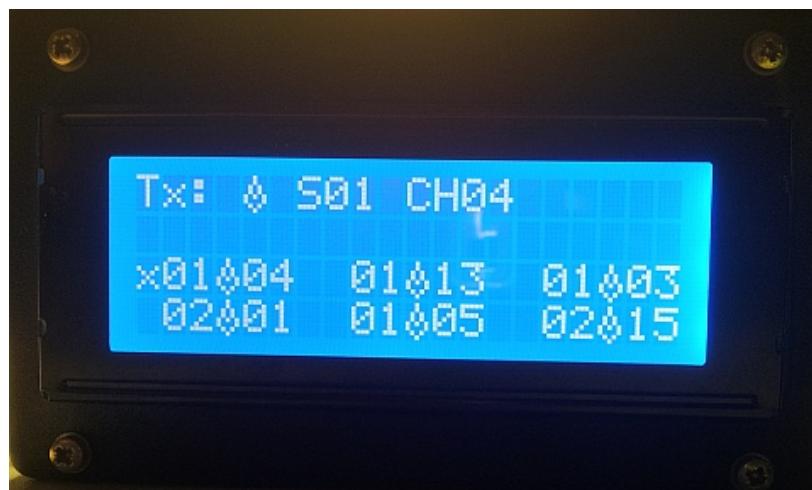


Abbildung 1.3: LCD während der Show

## 1.6 Schalter an den Zündboxen

Die Zündboxen verfügen, wie in Abbildung 1.4 zu erkennen, über zwei Schalter, einen schwarzen Wippschalter zum Ein- und Ausschalten der Energieversorgung sowie einen Schlüsselschalter, um die Zündbox „scharf“ zu schalten.

Die Scharfschaltung durch den Schlüsselschalter geschieht auf die Weise, dass durch eine Zustandsabfrage vor der Zündung letztere nur ausgeführt wird, wenn das Schloss auf den grünen Punkt am Gehäuse zeigt. Befindet sich der Schlüssel in waagrechter Stellung und zeigt auf den roten Punkt, so ist der Schalter geöffnet und Zündbefehle werden von der Box ignoriert. Für das Scharfschalten der Box ist der Anwender selbst verantwortlich.



Abbildung 1.4: Schalter und serielle Schnittstelle an der Zündbox

Ob Boxen scharf geschaltet sind, ist an der Box – wie in Abschnitt 1.4 ausgeführt – durch das Dauerleuchten der roten Status-LED erkennbar, kann aber auch durch eine Identifizierungsabfrage ausgelesen werden (siehe Abschnitt 3.1.2).

## 2 Vorbereitung des PCs

Zur Kommunikation mit einem Computer verfügen alle Devices über eine serielle Schnittstelle bzw. ab der dritten Generation über einen USB-Anschluss. *Pyro Ignition Control* sollte ohnehin auf dem Rechner installiert sein, für die serielle Kommunikation gibt es für Windows zudem zahlreiche Terminalprogramme.

### 2.1 Installation eines USB-RS232-Adapters

Zunächst steht man allerdings in der Regel vor dem Problem, dass zwar die Devices eine serielle Schnittstelle besitzen, moderne Rechner aber nicht mehr mit dem früher standardmäßig verbauten 9-poligen Sub-D-Stecker der RS232-Schnittstelle ausgestattet sind. Diese wurden seit dem Ende der 1990er-Jahre von den USB-Schnittstellen verdrängt. Sollte wider Erwarten am einzusetzenden Rechner ein derartiger Anschluss vorhanden sein, können die nächsten Absätze übersprungen und der COM-Port direkt im Gerätemanager anhand von Tabelle 2.1 konfiguriert werden. Wer nur über USB-Ports verfügt, lese unmittelbar weiter.

Weil in vielen Bereichen noch immer auf RS232 zurückgegriffen wird, existieren Adapterkabel wie in Abbildung 1.2a mit USB-Anschluss für den Rechner und einem 9-poligen RS232-Stecker für den Anschluss an der Peripherie, also die Devices von *El Fueradore*. In diesen Adapterkabeln ist ein Chip verbaut, um die Signalumsetzung von USB auf RS232 und umgekehrt zu bewerkstelligen. Übliche verwendete Chips sind der CH340<sup>1</sup>, welcher sich in vielen über eBay aus China angebotenen Modellen befindet, der Prolific PL2303<sup>2</sup> in verschiedenen Versionen oder – bei edleren und somit auch teureren Varianten – der uneingeschränkt zu empfehlende FTDI232, mit welchem die in den folgenden Abschnitten beschriebenen Probleme nicht auftreten sollten.

Das Plug-and-Play-Traumszenario, dass sich der Adapter bei der Verbindung des USB-Steckers mit dem Rechner automatisch korrekt installiert, tritt gerade bei den günstigen Adaptern leider nur sehr selten ein. Die in den Fußnoten verlinkten Treiber sollten, sofern die automatische Treiberinstallation von Windows versagt, ihren Dienst tun, müssen allerdings teilweise mit sanfter Gewalt installiert werden. Hat man die Installation erfolgreich absolviert, sollte bei angeschlossenem Adapterkabel ein neuer Eintrag in der Art von Abbildung 2.1 im Gerätemanager auftauchen.

<sup>1</sup>Treiber CH340/341: <http://wch.cn/download/list.asp?id=5>

<sup>2</sup>Falls die automatische Treiberinstallation unter Windows fehlschlägt, funktioniert – mit zeitweiligen Aussetzern – oft der Treiber unter: [http://www.cartft.com/support/drivers/TFT/tftdrivers/GPS/PL2303\\_Prolific\\_GPS\\_1013\\_20090319.zip](http://www.cartft.com/support/drivers/TFT/tftdrivers/GPS/PL2303_Prolific_GPS_1013_20090319.zip)



Abbildung 2.1: Eintrag des USB-RS232-Adapters im Gerätetandler

Weil viele Chips dazu tendieren, sich beim Anschluss an immer wieder andere USB-Ports neu zu installieren bzw. eine andere COM-Port-Nummer anzunehmen, wird empfohlen, für den USB-RS232-Adapter stets denselben USB-Steckplatz zu nutzen.

Durch Doppelklick auf den Eintrag und den Reiter Anschlusseinstellungen kann die nun vorhandene serielle Schnittstelle unter Windows konfiguriert werden. Um Kompatibilität mit *Pyro Ignition Control* zu gewährleisten, ist das Hauptfenster nach Tabelle 2.1 zu konfigurieren.

Unter der Schaltfläche „Erweitert“ kann man zudem die Puffer ausschalten, was aber nicht zwingend notwendig ist und die Funktionsweise normalerweise weder positiv noch negativ beeinflusst, sowie die Portnummer für den neu geschaffenen COM-Port einstellen.

Bits pro Sekunde	9600
Datenbits	8
Parität	keine
Stoppbits	1
Flusssteuerung	keine

Tabelle 2.1: Konfiguration der seriellen Schnittstelle

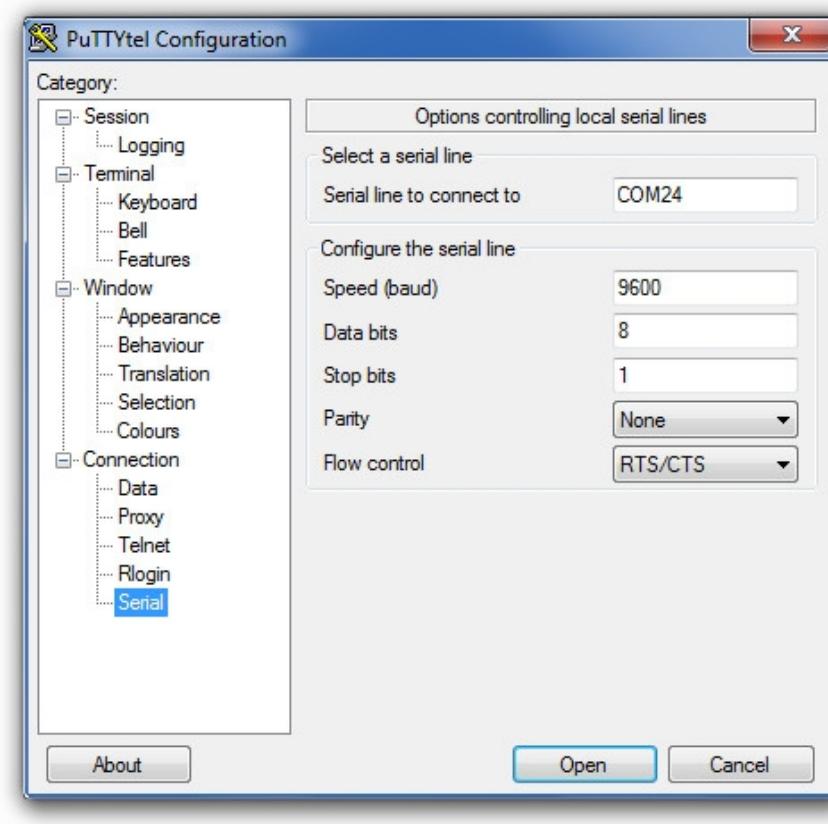


Abbildung 2.2: Einstellungen für Puttytel

## 2.2 Einrichtung des Terminalprogramms

Hardware- und treiberseitig steht einer erfolgreichen Kommunikation von Rechner und Devices nun nichts mehr im Wege, für eine komfortable Unterhaltung außerhalb von *Pyro Ignition Control* fehlt aber noch die entsprechende Software. Empfohlen wird die Verwendung des kostenlosen Programms *Puttytel*<sup>3</sup>, mit welchem auch die im Rahmen dieser Anleitung gezeigten Beispiele durchgeführt werden. Es besteht nur aus einer einzigen ausführbaren Datei.

**Puttytel** kann per Doppelklick gestartet werden, woraufhin man zu einem Startbildschirm, der Kategorie „Session“ gelangt. Man wählt in der linken Spalte unten links „Serial“ und stellt die Parameter – analog zur Konfiguration des COM-Ports nach Tabelle 2.1 – wie in Abbildung 2.2 ein. Anschließend muss man in der linken Spalte zurück auf „Session“ gehen und dort im rechten Teil des Fensters „Serial“ als „Connection Type“ wählen, ehe man die serielle Verbindung per Klick auf „Open“ starten kann. Vor dem Start der Verbindung sollte man zudem das verbundene Device mit Strom versorgen, damit dieses seine Kommunikationsschnittstelle vor Beginn des Datenaustauschs initialisieren kann.

<sup>3</sup>Herunterzuladen unter: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Um bei **Puttytel** nicht immer alle Einstellungen per Hand vornehmen zu müssen, bietet sich an, unter Windows eine Verknüpfung auf **puttytel.exe** zu erstellen und in den Verknüpfungseigenschaften als Ziel anzugeben:

```
"c:\programme\puttytel\puttytel.exe" -serial com24 -sercfg 9600,8,1,n,R
```

Die ohne Leerzeichen auf „com“ folgende Zahl ist natürlich entsprechend dem verwendeten seriellen Anschluss (COM5, COM37, ...) anzupassen.

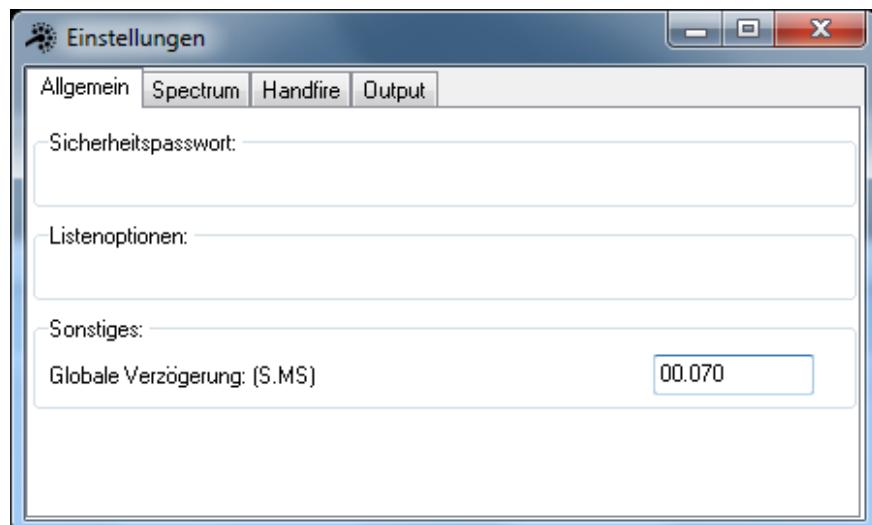
## 2.3 Einrichtung von *Pyro Ignition Control*

Um eine reibungslose Kommunikation zwischen *Pyro Ignition Control* und *El Fueradoro* sicherzustellen, muss in *Pyro Ignition Control* als wesentliche Einstellung unter dem Menüpunkt „Einstellungen → Optionen“ im Reiter „Output“ – gezeigt in Abbildung 2.3a – der richtige COM-Port eingestellt werden. Über „Einstellungen → Connect“ wird die serielle Verbindung aufgebaut und in der untersten Leiste angezeigt, ob der Verbindungsaufbau erfolgreich war.

Zudem sollte im Reiter „Allgemein“ die globale Verzögerung erfahrungsgemäß, wie Abbildung 2.3b zeigt, auf etwa 0,07 s eingestellt werden. Dies ist die Zeit, die aufgrund von Datenübertragungen und Rechenvorgängen zwischen dem Beginn des Sendens des Befehls vom PC zum Transmitter und dem Zünden des Kanals an der Zündbox vergeht.

Als minimale Zeitdauer zwischen zwei Zündungen sollte 100 ms nicht unterschritten werden, das Scharfschalten vor Beginn der Show ist ebenso nicht zu vergessen wie das Scharfschalten des Transmitters und sämtlicher Zündboxen!

**Anmerkung: Eine serielle Verbindung zu einem Device kann immer nur durch einen einzigen Client (Puttytel, GUI, Pyro Ignition Control, Firmware-Updater) bestehen. Man muss also immer die bestehende Verbindung trennen, bevor man mit einem anderen Programm eine neue aufbauen kann.**

(a) Einstellung des COM-Ports in *Pyro Ignition Control*(b) Einstellung des Global Delay in *Pyro Ignition Control*Abbildung 2.3: Einstellungen in *Pyro Ignition Control*



## 3 Kommunikation zwischen PC und Devices

In diesem Abschnitt wird die Systemüberwachung bzw. -konfiguration über die serielle Schnittstelle mittels **Puttytel** oder die *El Fueradoro*-GUI behandelt.

Auf die Kommunikation zwischen *Pyro Ignition Control* und der Transmitterbox wird an dieser Stelle nicht detailliert eingegangen, da hier – wenn alle Einstellungen wie in Abschnitt 2.3 erläutert getroffen wurden – alles quasi-automatisch und ohne Zutun des Benutzers stattfindet. Es muss dafür aber unbedingt sichergestellt sein, dass das von *Pyro Ignition Control* angesprochene Device scharfgeschaltet sein muss. Dies geschieht bei Zündboxen via Schlüsselschalter, bei Transmittern über den Terminalbefehl „arm“.

Im Abschnitt 3.3 ab Seite 35 wird das Aktualisieren der Firmware mittels Firmware-Updater erklärt.

### 3.1 Puttytel

#### 3.1.1 Befehlsübersicht

Hat man mittels **Puttytel** eine Verbindung zwischen einem Device und dem PC aufbauen können, sieht man vor sich zunächst nur einen schwarzen Bildschirm. Um nun mit dem Device kommunizieren zu können, existieren einige Befehle gemäß Tabelle 3.1.

Diese können, sofern **Puttytel** die aktive Anwendung ist, direkt über die PC-Tastatur eingegeben werden und sollten zur unmittelbaren Ausführung mit Druck auf die Taste *ENTER* abgeschlossen werden. Sobald das erste Zeichen eingegeben wurde, leuchtet die gelbe Status-LED am Device. Unbekannte Befehle werden ignoriert, sämtliche Buchstaben als Kleinbuchstaben interpretiert. Korrekturen sind unter Verwendung der *BACKSPACE*-Taste möglich.

Aufgrund der eingebauten Timeout-Funktion, welche ein Hängenbleiben des Programms während einer Show verhindern soll, bricht die Firmware die Eingabe ab, wenn zwischen der Eingabe der einzelnen Buchstaben mehr als 3 s vergehen. Lässt man diese Zeit verstreichen, wird automatisch ein Drücken der *ENTER*-Taste übermittelt, die Befehlseingabe also abgeschlossen und das Device ist unmittelbar bereit, einen neuen Befehl aufzunehmen. Wird also nach Eingabe eines gültigen Befehls die *ENTER*-Taste nicht gedrückt, wird der Befehl durch den Timeout dennoch ausgelöst. Möchte

<b>Befehl</b>	<b>Wirkung</b>
arm	Schaltet den Transmitter scharf, um Zündbefehle senden zu können
disarm	Schaltet den Transmitter unscharf, Zündbefehle werden nicht mehr gesendet
conf	Startet das Konfigurationsprogramm zur lokalen Zuweisung von Unique- und Slave-ID
remote	Startet das Konfigurationsprogramm zur ferngesteuerten Zuweisung von Unique- und Slave-ID
list	Zeigt die Systemübersicht (Zuweisung Unique- und Slave-ID, Batteriespannung jeder Box, Scharfschaltungsstatus Temperatur, RSSI, Anzahl Boxen je Slave-ID)
send	Startet das Menü zur manuellen Eingabe einer Anweisung ans Funkmodul (Zündbefehl, Identifizierungsaufforderung oder Temperaturmessung)
fire	Führt zu einer Eingabemaske, in die Slave-ID und Kanal für die Zündung einzugeben sind
ident	Sendet eine Identifizierungsaufforderung an alle anderen Devices
temp	Gibt über die serielle Schnittstelle die Temperatur aus und fordert alle anderen Devices ebenfalls zur Temperaturmessung auf. Zum Auslesen der neu gemessenen Temperaturen muss dann eine Identifizierungsanfrage geschickt werden
rfm	Erlaubt unmittelbaren Zugriff auf das Funkmodul durch Eingabe einer 16-Bit-Hexadezimalzahl, um Registerwerte auszulesen oder neu zu setzen
aeskey	Schlüssel für die Funkübertragung auslesen und neu setzen
orders	Gibt letztes gesendetes und empfangenes Pattern auf LCD aus
cls	Löscht den Terminal-Bildschirm
kill	Löst einen Neustart des Device aus

Tabelle 3.1: Kommandos zur Konfiguration über die serielle Schnittstelle

man dies vermeiden, sollte man den Befehl vor Ausführung durch Eingabe weiterer Zeichen ungültig machen oder durch Entfernen aller Zeichen mittels *BACKSPACE* löschen.

Von den in Tabelle 3.1 aufgeführten Befehlen funktioniert lediglich „orders“ nicht bei allen Devices, sondern setzt voraus, dass das angeschlossene Device ein Transmitter ist.

### 3.1.2 Konfiguration

#### Lokal

Mit „conf“ gelangt man ins Konfigurationsmenü für die lokale Konfiguration der IDs, dessen Ablauf beispielhaft in Abbildung 3.1 gezeigt ist. Hier sind die beiden wichtigsten Parameter jeder Zündbox, die Unique-ID und die Slave-ID, aufgeführt, die der Benutzer nach eigenen Bedürfnissen vergeben kann.



Abbildung 3.1: Ablauf des Konfigurationsprogramms bei Verbindung mit einer Zündbox

Die **Unique-ID** dient der Identifikation jeder einzelnen Zündbox im Funksystem. Jeder verwendeten Zündbox muss daher, um die Funktion von *El Fueradoro* gewährleisten zu können, eine andere Unique-ID im Bereich von 01-30 (zweistellige Eingabe!) eindeutig zugeteilt werden, d. h. jede Box hat eine unterschiedliche Unique-ID bzw. jede Unique-ID gehört zu genau einer Zündbox.

Die **Slave-ID** entscheidet, auf welche Zündbefehle eine Zündbox reagiert. Sollen also zwei oder mehr Boxen stets zur selben Zeit denselben Kanal zünden, kann ihnen einfach die gleiche Slave-ID zugewiesen werden.

Die Null als Unique- und Slave-ID identifiziert ein Device als Transmitterbox. Die Software für Transmitter und die ersten beiden Zündboxgenerationen erkennt dabei automatisch, ob es sich beim angeschlossenen Device um einen Transmitter oder eine Zündbox handelt<sup>4</sup> und weist Transmittern unmittelbar beim Hochfahren „0“ als Unique- und Slave-ID zu. Ein vom Aufbau her als Zündbox ausgeführtes Device muss immer von Null verschiedene IDs besitzen.

Die Zuweisung von Unique- und Slave-ID vom Startbildschirm des Konfigurationsprogramms aus geschieht, indem man den Anweisungen auf dem Bildschirm folgt. Die Eingabe von „I“ bzw. „i“ (Groß- oder Kleinschreibung spielt keine Rolle) erlaubt eine Änderung von Unique- und Slave-ID einer Zündbox. Die neue Unique- bzw. Slave-ID muss stets zweistellig ohne Bestätigung durch *ENTER* oder eine andere Taste eingegeben werden. Beide IDs können im Bereich von 01-30 liegen.

Möchte man beispielsweise die Unique-ID 5 und die Slave-ID 12 zuweisen, muss man nach Anzeige des Startbildschirms zunächst „i“ und anschließend „05“ und „12“ eingeben. Will man eine der beiden IDs beibehalten und nur die andere ändern, kann die Änderung durch Drücken von *ENTER* übersprungen werden. Die zugewiesenen IDs werden an drei Stellen im internen Speicher mit Prüfsummen hinterlegt und bleiben sowohl nach dem Ausschalten als auch nach einem Firmwareupdate erhalten. Eine Änderung mindestens einer der IDs führt zu einem sofortigen Neustart der Zündbox.

<sup>4</sup>Wie diese automatische Erkennung funktioniert, ist im Abschnitt 8 ab Seite 79 beschrieben

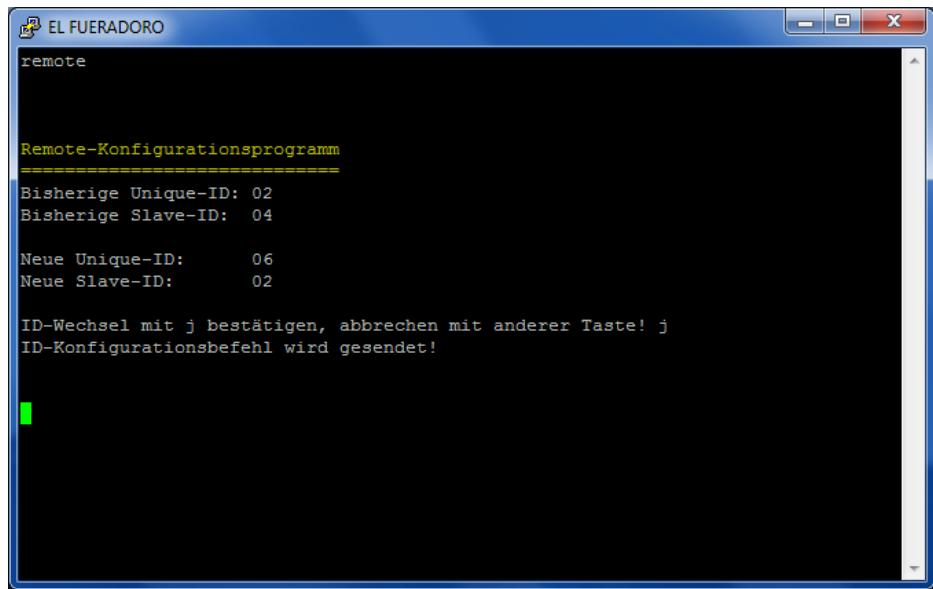


Abbildung 3.2: Beispiel einer ferngesteuerten ID-Zuweisung

Unabhängig davon, dass eine Zündbox nicht als Transmitter konfiguriert werden, also nicht Unique- und Slave-ID besitzen kann, kann sie theoretisch trotzdem zur Steuerung und Koordinierung eines Netzes und einer Show eingesetzt werden, indem man sie über die serielle Schnittstelle mit dem PC verbindet. Bis auf die Darstellung am LCD erfüllt sie dieselben Aufgaben wie ein Transmitter und kann auch parallel noch als Zündbox fungieren. Auf entsprechenden Sicherheitsabstand zu Lebewesen und sensibler Technik ist dabei selbstverständlich zu achten!

### Ferngesteuert

Über den Befehl „remote“ gelangt man ins Konfigurationsprogramm zur ferngesteuerten Vergabe von Unique- und Slave-ID. Der Programmablauf ist beispielhaft in Abbildung 3.2 gezeigt: Die Eingabe der alten und neuen IDs erfolgt analog zur Eingabe bei lokaler Konfiguration, am Ende muss die Änderung noch bestätigt werden. Werden als alte Unique- und Slave-ID die Daten der verbundenen Box eingegeben, so werden deren Kennzahlen wie bei einer lokalen Konfiguration geändert und kein weiterer Befehl gesendet.

Durch „remote“ und die Eingabe von Unique- und Slave-ID einer nicht per Kabel verbundenen Box ist es möglich, die IDs einer eingeschalteten Zündbox per Funk zu ändern. Voraussetzung ist dabei, dass die angesprochene Zündbox nicht scharf geschaltet ist.

Der Anwender hat selbst darauf zu achten, durch ein ferngesteuertes Update nicht einem Device eine bereits vergebene Unique-ID zuzuweisen! Falls dies dennoch geschieht, ist das weitere Vorgehen davon abhängig, ob die Devices auch die gleiche Slave-ID besitzen oder nicht. Sind die Slave-IDs nicht identisch, so kann durch einen weiteren „remote“-Befehl die Unique-ID-Zuweisung geändert werden.

Bei identischen Slave-IDs funktioniert dies nicht, da stets alle Devices auf den Änderungsbefehl in gleicher Weise reagieren würden. Hier müssen daher alle Devices mit identischen IDs bis auf eines ausgeschaltet werden, dem man dann neue IDs zuweisen kann. Nun kann dann jeweils ein weiteres Device eingeschaltet und seine IDs neu gesetzt werden, bis wieder alle unterschiedliche Unique-IDs besitzen.

### 3.1.3 Systemübersicht

Mit „list“ ist es möglich, sich die Systemübersicht entsprechend Abbildung 3.3 anzeigen zu lassen. Es werden zwei Tabellen ausgegeben, wobei die obere nach Unique-ID geordnet angezeigt:

1. Slave-ID, welcher der Unique-ID zugewiesen ist.
2. Spannung, welche die Batterie der Box mit der entsprechenden Unique-ID liefert.
3. Scharfschaltungsstatus der Box mit der jeweiligen Unique-ID: (j)a (=scharf) oder (n)ein (=nicht scharf).
4. Temperatur im Inneren der Box, sofern die Box über einen eingebauten Temperatursensor verfügt, ansonsten wird „n.a.“ (not available) angezeigt.
5. Stärke des von der Box empfangenen Antwortsignals (RSSI = Received Signal Strength Indicator) in dBm. Je größer der Wert ist – bei negativen Werten also umso näher er bei 0 liegt, umso besser und umso weniger störanfällig ist die Verbindung zwischen den Devices. Die theoretische Empfangsgrenze liegt bei etwa -96 dBm.

Die untere Tabelle listet auf, wie viele Boxen mit der entsprechenden Slave-ID derzeit aktiv sind.

Zwischen den beiden Tabellen wird die Anzahl der fehlerhaften IDs aufgelistet. Dies kann entweder auf doppelte Zuweisung von Unique-IDs oder Fehler beim Auslesen der IDs (fehlerhafte Prüfsummen) zurückzuführen sein. Für normalen Betrieb sollte dieser Wert stets 0 betragen.

Der dargestellte Zustand entspricht den empfangenen Parametern nach der letzten Identifikationsaufforderung bzw. nach dem Einschalten der Zündbox. Für eine möglichst aktuelle Liste sollte also vor dem Aufruf von „list“, wie im Abschnitt 3.1.4 beschrieben, eine Identifikationsaufforderung gesendet werden.

### 3.1.4 Manuelles Senden

Zu Testzwecken oder um die Systemübersicht zu aktualisieren, können mittels „send“ Zündbefehle und die Aufforderung zur Identifizierung oder Temperaturmessung manuell versendet werden. Nach Eingabe von „send“ muss dies mit „f“ (=fire), „i“ (=identify) oder „t“ (=temperature) ausgewählt

```

EL FUERADORO
Systemübersicht
=====

Unique-ID: Slave-ID, Batteriespannung (V), Scharf?, Temperatur (°C), RSSI (dBm)
01: 01, 12.8, n, 21, -33 02: 01, 12.4, n, 21, -24 03: 02, 13.0, n, 21, -35
04: ---, ---, -, ---, --- 05: ---, ---, -, ---, --- 06: ---, ---, -, ---, ---
07: ---, ---, -, ---, --- 08: ---, ---, -, ---, --- 09: ---, ---, -, ---, ---
10: ---, ---, -, ---, --- 11: ---, ---, -, ---, --- 12: ---, ---, -, ---, ---
13: ---, ---, -, ---, --- 14: ---, ---, -, ---, --- 15: ---, ---, -, ---, ---
16: ---, ---, -, ---, --- 17: ---, ---, -, ---, --- 18: ---, ---, -, ---, ---
19: ---, ---, -, ---, --- 20: ---, ---, -, ---, --- 21: ---, ---, -, ---, ---
22: ---, ---, -, ---, --- 23: ---, ---, -, ---, --- 24: ---, ---, -, ---, ---
25: ---, ---, -, ---, --- 26: ---, ---, -, ---, --- 27: ---, ---, -, ---, ---
28: ---, ---, -, ---, --- 29: ---, ---, -, ---, --- 30: ---, ---, -, ---, ---

Fehlerhafte/doppelte IDs: 0

Slave-ID: Anzahl Boxen
01: 2 02: 1 03: ---
04: --- 05: --- 06: ---
07: --- 08: --- 09: ---
10: --- 11: --- 12: ---
13: --- 14: --- 15: ---
16: --- 17: --- 18: ---
19: --- 20: --- 21: ---
22: --- 23: --- 24: ---
25: --- 26: --- 27: ---
28: --- 29: --- 30: ---

```

Abbildung 3.3: Systemübersicht

werden. Wählt man „i“ oder „t“ ist keine weitere Eingabe nötig, bei „f“ müssen anschließend noch Slave-ID und Kanal jeweils zweistellig eingegeben werden. Statt „send“ und den entsprechenden Buchstaben anzugeben, können auch die direkten Befehle „fire“, „ident“ und „temp“ verwendet werden.

Jede andere Angabe als „f“, „i“ oder „t“ beendet den Modus ohne irgendetwas zu senden. Den-selben Effekt hat die Eingabe einer Slave-ID oder Kanalnummer außerhalb der jeweils zulässigen Zahlenbereiche.

### 3.1.5 Funkmodul-Zugriff

Die zwingend für den Betrieb von *El Fueradoro* notwendigen Zugriffe auf das Funkmodul werden von der Software automatisch getätig, so dass diese Funktion in der Regel nicht gebraucht wird. Dennoch ist es möglich das verwendete Funkmodul RFM69CW bzw. RFM69HCW<sup>5</sup> unmittelbar über das Terminalprogramm anzusprechen, um Werte aus den Registern zu lesen oder die Register für die aktuelle Sitzung neu zu beschreiben. Beim nächsten Neustart des Devices werden stets die Standardeinstellungen wiederhergestellt.

Nach Eingabe von „rfm“ und Bestätigung mit *ENTER* erscheint eine Aufforderung zur Befehlseingabe. Diese hat im Hexadezimalformat als 16-Bit-Wert zu erfolgen, d. h. vierstellig mit den zulässigen Zeichen 0-9 und A-F bzw. a-f. Jedes eingegebene Zeichen symbolisiert dabei vier Bits, die Umrechnung

<sup>5</sup>Link zum Datenblatt auf Seite 77

Hexadezimal	0	1	2	3	4	5	6	7
Dezimal	0	1	2	3	4	5	6	7
Binär	0000	0001	0010	0011	0100	0101	0110	0111
Hexadezimal	8	9	A	B	C	D	E	F
Dezimal	8	9	10	11	12	13	14	15
Binär	1000	1001	1010	1011	1100	1101	1110	1111

Tabelle 3.2: Umrechnung Hexadezimal-, Dezimal- und Binärwerte

Bit	w/r	r6	r5	r4	r3	r2	r1	r0	d7	d6	d5	d4	d3	d2	d1	d0			
Wert	8	4	2	1		8	4	2	1		8	4	2	1		8	4	2	1
Zeichen		Zeichen 1				Zeichen 2				Zeichen 3				Zeichen 4					
<hr/>																			
w/r      Schreib- oder Lesezugriff (0 = lesen, 1 = schreiben) r6 ... r0   Registeradresse d7 ... d0   Zu schreibender Registerwert (beliebig falls w/r = 0)																			

Tabelle 3.3: Struktur des RFM69-Befehls

ist in Tabelle 3.2 gezeigt.

Die Bedeutung der Eingabe für das Funkmodul ist in Tabelle 3.3 illustriert. Hierbei sollte auch klar werden, wie sich die Werte für die Zeichen 1-4 zusammensetzen. Ist ein Bit gesetzt, muss die entsprechende Zahl (8, 4, 2, 1) zum Zeichenwert addiert werden, so dass sich bei vier gesetzten Bits als Maximalwert 15 ergibt, ist nur das oberste Bit gesetzt, lautet der Wert 8, ist nur das unterste gesetzt 1, sind die beiden mittleren Bits gesetzt 6, usw.

Es ist zu erkennen, dass das erste einzugebende Zeichen sowohl das Schreiben/Lesen-Bit enthält als auch die obersten drei Bit der Registeradresse. Die acht Datenbits sind lediglich für einen Schreibbefehl relevant, bei einem Lesezugriff kann als drittes und viertes Zeichen ohne Konsequenzen ein beliebiger Hexadezimalwert im Bereich von 0x00 bis 0xFF übertragen werden.

### Auslesen der eingestellten Sendeleistung

Zur Veranschaulichung soll hier die Abfrage der aktuell eingestellten Sendeleistung und eine anschließende Änderung derselben mit dem Funkmodul RFM69CW simuliert werden: Aus dem Datenblatt, dem die Bedeutungen aller Registeradressen und ihrer acht Registerbits zu entnehmen sind, kann die Registeradresse 0x11 als diejenige identifiziert werden, in der die Informationen zur Sendeleistung hinterlegt sind. Um nun den aktuellen Wert auszulesen, gibt man – wie in Abbildung 3.4 gezeigt – im

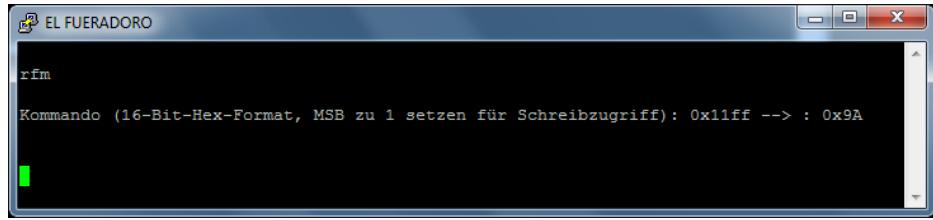


Abbildung 3.4: Auslesen der gesetzten Sendeleistung

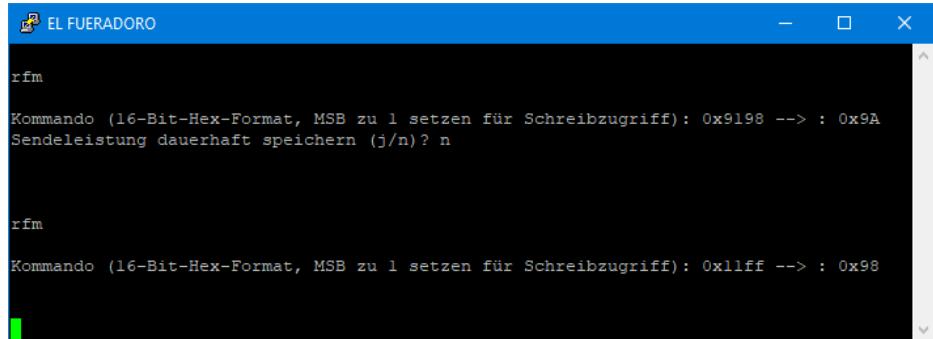


Abbildung 3.5: Setzen der Sendeleistung mit anschließendem Auslesen

Terminalprogramm „rfm“ gefolgt von *ENTER* ein und anschließend die Zeichenfolge „11FF“, wobei die beiden hinteren Stellen wie erwähnt keine Rolle spielen.

Das Modul antwortet nun mit einem 8-Bit-Wert, z. B. mit dem Wert 0x9A, welcher dem Binärwert 10011010 entspricht, dessen Bedeutung dem Datenblatt entnommen werden kann: Das oberste Bit signalisiert, dass die Verstärkerstufe PA0 aktiv ist, die beiden folgenden Bits sind 0, da PA1 und PA2 in der Variante RFM69CW nicht genutzt werden können. Die unteren fünf Bits schließlich stehen für die eingestellte Sendeleistung, wobei man vom aus den fünf Bits berechneten Wert noch 18 abziehen muss, um die Sendeleistung in dBm zu erhalten. Gesetzt sind die Bits 4, 3 und 1, was dem Wert 26 (=  $2^4 + 2^3 + 2^1$ ) entspricht, daraus resultiert eine eingestellte Sendeleistung von 8 dBm.

### Setzen der Sendeleistung

Will man die Sendeleistung nun auf 6 dBm anpassen, muss also ein Wert von 24 für die Ausgangsleistungs-Bits gesetzt werden, dazu natürlich auch das oberste Bit für den PA0. Als Wert für die Registerbits ergibt sich damit  $2^7 + 2^4 + 2^3 = 0x98$ . Die Registeradresse bleibt gleich, jedoch muss dem Modul mitgeteilt werden, dass es sich um einen Schreibzugriff handelt, weshalb die erste Stelle um den Wert 8 erhöht werden muss. Um nun den neuen Wert von 6 dBm einzuschreiben, gibt man im Terminalprogramm – wie in Abbildung 3.5 gezeigt – „rfm“ gefolgt von *ENTER* ein und anschließend die Zeichenfolge „9198“.

Als Antwort erhält man vom Modul den Registerwert von VOR dem Schreibzugriff, im Beispiel also

den Wert 0x9A. Ein nochmaliges Auslesen des Registers, wie im Abschnitt 3.1.5 beschrieben, sollte nun den eben eingeschriebenen Wert 0x98 zurückgeben. Beim nächsten Neustart des Device wird der Standardwert wiederhergestellt.

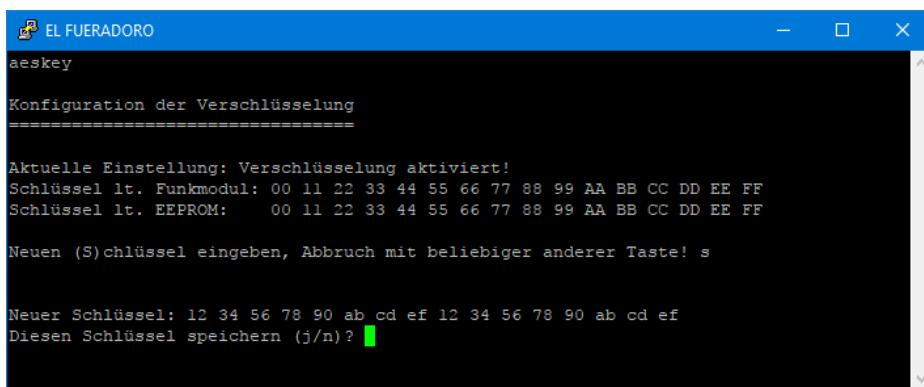
### 3.1.6 AES-Verschlüsselung

Zum sicheren Betrieb von *El Fueradoro* werden die drahtlos zu übermittelnden Daten zwischen Sender und Empfänger verschlüsselt. Hierfür muss in den am Funkverkehr beteiligten Modulen ein Schlüssel mit einer Länge von 16 Bytes (128 Bit) hinterlegt werden, der für die Ver- und Entschlüsselung der Datenpakete genutzt wird. Dieser Schlüssel ist im internen Speicher des Mikrocontrollers hinterlegt und bleibt auch beim Trennen der Stromversorgung erhalten. Selbstredend muss in allen Controllern derselbe Schlüssel hinterlegt sein, damit diese miteinander kommunizieren können.

Nach Eingabe von „aeskey“ im Terminalfenster kann der aktuell eingestellte Schlüssel eingesehen werden. Hierzu wird der Schlüssel einmal aus dem Speicher des Controllers ausgelesen und einmal aus den Einstellungen des Funkmoduls. Diese beiden Zeilen sollten identische Werte anzeigen.

Jede Taste außer „s“ bringt den Benutzer anschließend ins Hauptmenü zurück.

Möchte man den Schlüssel ändern, muss die Taste „s“ gedrückt und anschließend der komplette neue Schlüssel im hexadezimalen Zahlenformat eingegeben werden, wie beim Funkmodul-Zugriff sind also nur die Zeichen 0-9 und A-F bzw. a-f zulässig. Wurde ein gültiger Schlüssel eingegeben, erfolgt noch eine finale Nachfrage, ob dieser neue Schlüssel gespeichert werden soll, welche man mit „j“ oder „n“ beantworten kann. Abbildung 3.6 zeigt das Prozedere im Terminalfenster.



```
EL FUERADORO
aeskey
Konfiguration der Verschlüsselung
=====
Aktuelle Einstellung: Verschlüsselung aktiviert!
Schlüssel lt. Funkmodul: 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF
Schlüssel lt. EEPROM:      00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF

Neuen (S)chlüssel eingeben, Abbruch mit beliebiger anderer Taste! s

Neuer Schlüssel: 12 34 56 78 90 ab cd ef 12 34 56 78 90 ab cd ef
Diesen Schlüssel speichern (j/n)?
```

Abbildung 3.6: Auslesen und Neusetzen des AES-Schlüssels

### 3.2 GUI

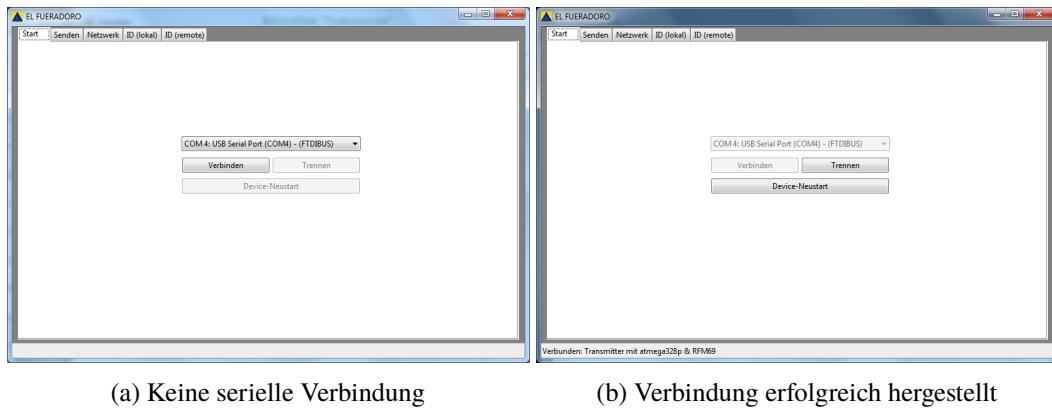


Abbildung 3.7: Startbildschirm der GUI

Die GUI stellt eine graphische Oberfläche zur Implementierung der **Puttytel**-Funktionen dar, indem über die serielle Schnittstelle ein- und ausgehende Daten geparsst und an den entsprechenden Stellen innerhalb der Oberfläche dargestellt werden. Die Software ist in die fünf Reiter „Start“, „Senden“, „Netzwerk“, „ID (lokal)“ und „ID (remote)“ unterteilt, in denen die entsprechenden Funktionen ausgeführt werden können.

Nach dem Start der Software sind alle Schaltflächen in allen Reitern mit Ausnahme der Auswahl der zur Verfügung stehenden COM-Ports und der Schaltfläche „Verbinden“ im Reiter „Start“ ausgegraut und deaktiviert, was in Abbildung 3.7a dargestellt ist. Nach Auswahl des entsprechenden Ports und Klick auf „Verbinden“, werden diese beiden Schaltflächen ausgegraut und deaktiviert. Im Gegenzug erscheint in der Fußleiste der Devicetyp mit verwendetem Controller und Funkmodul und die Schaltflächen „Trennen“ sowie „Device-Neustart“ werden im aktuellen Reiter aktiv (Abbildung 3.7b).

Nach Herstellung der Verbindung sind auch die sonstigen Schaltflächen aktiv, Abbildung 3.8 zeigt den Reiter „Senden“ im aktiven Zustand. Hier kann per Klick eine Identifizierungsaufforderung ver-

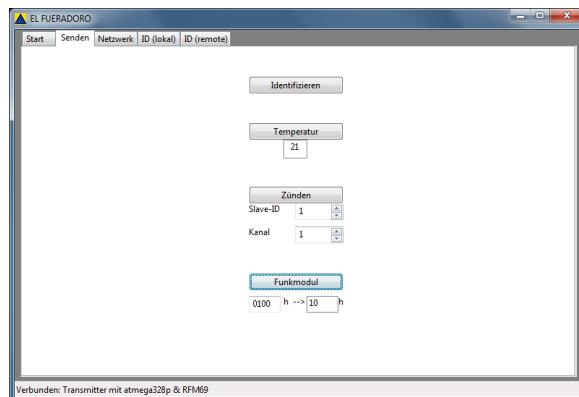


Abbildung 3.8: Sendebildschirm der GUI

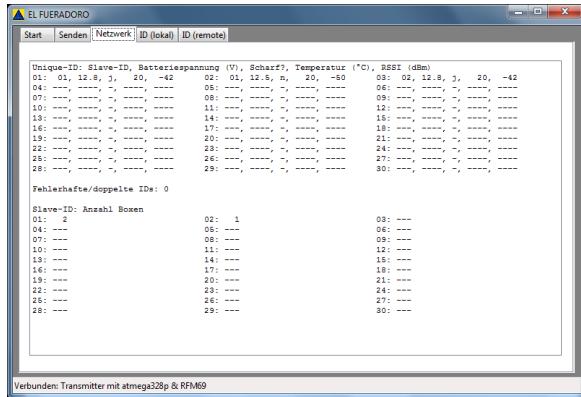


Abbildung 3.9: Netzwerkanzeige der GUI

schickt, eine Temperaturnmessung getriggert, ein Zündvorgang ausgelöst oder ein Befehl ans Funkmodul geschickt werden. Bei „Zünden“ und „Funkmodul“ ist darauf zu achten, dass beim Klick auf die Schaltfläche unmittelbar die aktuell in den Eingabefeldern stehenden Werte übertragen werden, man diese also vor dem Klicken anpassen muss. Bei Temperaturanfrage und Funkmodulzugriff wird im entsprechenden Anzeigefeld die Antwort des angeschlossenen Devices – also die Temperatur in Grad Celsius oder die Antwort des Funkmoduls auf den aktuellen Registerzugriff als 8-bit-Hexadezimalwert – ausgegeben. Das permanente Ändern der Sendeleistung ist in der GUI noch nicht implementiert.

Der Reiter „Netzwerk“ (Abbildung 3.9) dient einzig und allein der Darstellung der aktuell im System befindlichen Zündboxen mit ihren Parametern Unique-ID, Slave-ID, Batteriespannung, Scharfschaltungs-Status, Temperatur und gemessenem RSSI-Wert beim Empfang der Parameter.

Die Ansicht des Reiters „ID (lokale)“ ist, wie Abbildung 3.10 zeigt, abhängig vom angeschlossenen Devicetyp. Bei Transmittern sind die Schaltflächen deaktiviert, da hier keine Änderungen an den IDs vorgenommen werden können (Abbildung 3.10a). Bei Zündboxen werden bei Aufruf des Reiters die aktuellen IDs in die entsprechenden Felder kopiert, können verändert und die Änderungen dann durch Klick auf „Werte übernehmen“ übernommen werden (Abbildung 3.10b).

Im Reiter „ID (remote)“ können alte und neue ID-Kombination eingestellt werden. Durch Klick auf „Ausführen“ wird der Befehl zur ID-Änderung gesendet.

### 3.3 Firmwareupdate

**WICHTIG:** Aus Sicherheitsgründen darf kein Update durchgeführt werden, solange Anzünder mit dem Device verbunden sind, da sich die Devices im Falle eines Übertragungsfehlers völlig unvorhersehbar verhalten können.

### 3.3.1 Herunterladen

Die aktuelle Firmware kann aus dem Git-Repository

<https://github.com/fixxl/el-fueradoro>

mittels Git-Client oder direkt über den Link

<https://github.com/fixxl/el-fueradoro/archive/master.zip>

heruntergeladen werden.

Im Repository enthalten sind der komplette C-Quellcode, das AVR-Eclipse-Projekt, die kompilierten iHex-Dateien für Firmware und Bootloader, die nötigen Software-Tools zur Übertragung zwischen PC und Mikrocontroller sowie die vorliegende Anleitung.

### 3.3.2 Aktualisierung

*El Fueradoro* bietet die Möglichkeit, die Firmware via serielle Schnittstelle vom PC aus zu aktualisieren. Hierfür stehen das Kommandozeilentool *fwupdate.exe* oder eine Windows-Oberfläche namens *UpdateLoader*, welche beide zunächst einen Reset auslösen, um den Bootloader des Devices zu aktivieren, und anschließend die im iHex-Format vorliegende Firmware übertragen.

Es existieren für jede Kombination von Platinenversion (v1/v2, v3 mit 16 Kanälen und v3 mit 32 Kanälen) und Funkmodul (RFM69CW und RFM69HCW) Firmwareversionen. Tabelle 3.4 gibt einen Überblick über die Versionen, welche sich im gleichen Ordner wie *fwupdate.exe* und *UpdateLoader.exe* befinden.

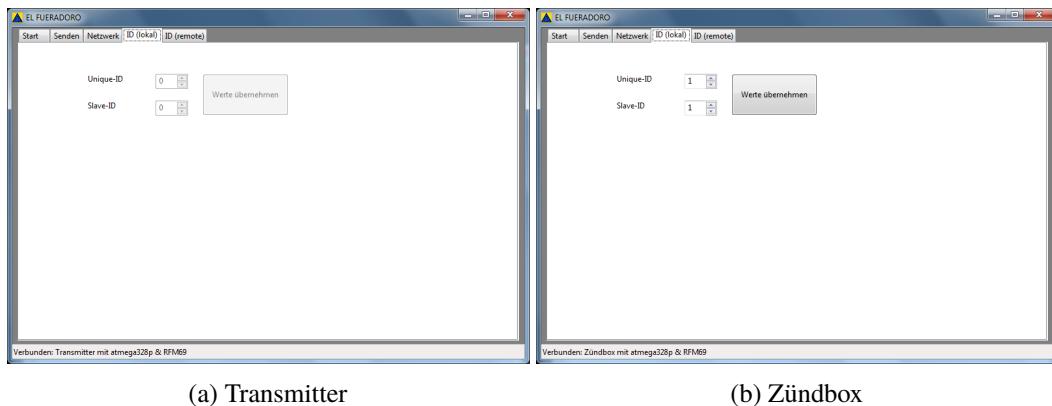


Abbildung 3.10: Lokale ID-Konfiguration

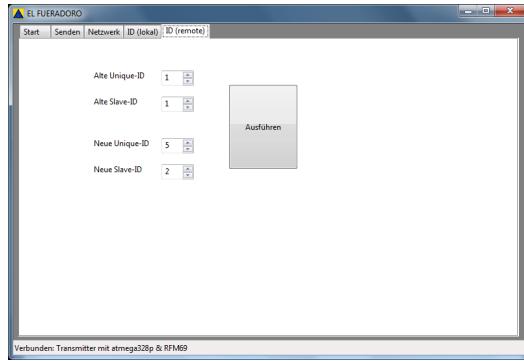


Abbildung 3.11: Remote-ID-Einstellung

Firmware	Platine	Kanäle	Funkmodul (Sendeleistung)
Pyro_atmega328p_RF69CW_16ch.hex	v1/v2	16	RFM69CW (13 dBm)
Pyro_atmega328p_RF69HCW_16ch.hex	v1/v2	16	RFM69HCW (20 dBm)
Pyro_v3_atmega328p_RF69CW_16ch.hex	v3	16	RFM69CW (13 dBm)
Pyro_v3_atmega328p_RF69CW_32ch.hex	v3	32	RFM69CW (13 dBm)
Pyro_v3_atmega328p_RF69HCW_16ch.hex	v3	16	RFM69HCW (20 dBm)
Pyro_v3_atmega328p_RF69HCW_32ch.hex	v3	32	RFM69HCW (20 dBm)

Tabelle 3.4: Verfügbare Firmware-Versionen

## Kommandozeile

*fwupdate.exe* muss über die Kommandozeile mit zwei Parametern gestartet werden, nämlich der Angabe der seriellen Schnittstelle und dem Namen der zu übertragenden Firmware-Datei. Um nun die neue Firmware zu übertragen, lautet das Kommando für ein Device am *COM4* mit dem *ATmega328P* (und dem Funkmodul *RFM69CW*):

```
fwupdate.exe /c4 /fm:Pyro_atmega328p_RF69.hex
```

oder für die Firmware mit höherer Sendeleistung

```
fwupdate.exe /c4 /fm:Pyro_atmega328p_RF69_HP.hex
```

Die Angabe der Dateiendung *.hex* kann hierbei – ebenso wie das *.exe* hinter *fwupdate* – auch weglassen werden, die Firmware-Datei muss jedoch zwingend auf *.hex* enden.

*fwupdate.exe* ist, sofern bereits eine korrekt funktionierende Firmware auf dem Device vorhanden ist, in der Lage, automatisiert zu ermitteln, welche Firmwaredatei die benötigte ist, das Kommando für ein Update über *COM4* lautet dann nur noch:

```
fwupdate.exe /c4 /fa
```

Der Updater führt nach Übertragung der Daten einen CRC-Check durch. Sollte dieser fehlschlagen, wurde die Firmware nicht korrekt übertragen. Dies kann zufällig passieren oder auf ein Hardwareproblem, welches in der Regel beim USB-RS232-Adapter liegt, zurückzuführen sein. Für den Fall eines CRC-Fehlers sollte die Firmware erneut übertragen werden. Bleibt das Update beim Punkt „COMx at 9600 baud:“ stehen, sollte die Stromversorgung des Device kurz unterbrochen und wieder aktiviert werden. Das Kabel für die serielle Verbindung bleibt währenddessen mit Device und PC verbunden.

## Windows-Oberfläche

Dank der Arbeit von Leo-Andres Hofmann<sup>6</sup> existiert die in Abbildung 3.12 dargestellte Windows-Oberfläche zur einfachen Übertragung der Firmware via serielle Schnittstelle, der **UpdateLoader**. Hier kann man aus der Liste der verfügbaren COM-Ports den entsprechenden wählen sowie die zu übertragende Datei einstellen (hier auf korrekte Version achten!) und die Dateiübertragung per Klick auf „Update starten“ durchführen.

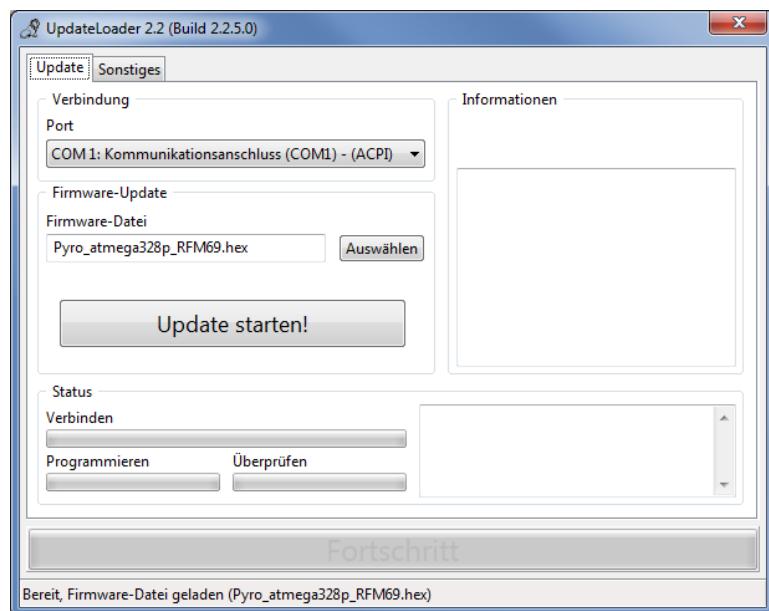


Abbildung 3.12: Benutzeroberfläche **UpdateLoader**

<sup>6</sup>Seine Homepage findet sich unter: <https://luani.de>

## 4 raspEasyFire

Als Alternative zur Kombination aus Laptop/PC und dem in den vorangegangenen Abschnitten beschriebenen Transmitter wurde in Zusammenarbeit mit einem Softwareentwickler ein Audio-Funksystem auf Basis des bekannten Einplatinencomputers „Raspberry Pi“ entwickelt. Dieser übernimmt die Kontrolle der Show inklusive der Musikwiedergabe.

Neben einem Raspberry Pi (unterstützt werden Stand 08-2020 Modelle bis 3B+) werden für den Aufbau eine speziell für die Verwendung mit *El Fueradoro* entwickelte Aufsteckplatine sowie ein Touchdisplay, eine USB-Soundkarte, zwei Taster (FIRE und ON-OFF), welche interne oder externe LEDs besitzen können, ein Schaltern und zwei weitere LEDs für die Signalisierung von Senden (TX) und Empfangen (RX) benötigt. Eine Einkaufsliste findet sich in Tabelle 10.2 ab Seite 89.

Ausgangspunkt für die Arbeiten mit *raspEasyFire* ist ein komplett aufgebautes System mit beschriebener SD-Karte, die notwendigen Schritte hierzu werden in den Abschnitten 12.2, 12.4.4 und 14 dargelegt. Eine mögliche Außenansicht mit den notwendigen Funktionseinheiten ist in Abbildung 4.1 gezeigt.

### 4.1 Erster Start

Beim ersten Hochfahren passt sich das System an die tatsächlichen lokalen Gegebenheiten an, insbesondere wird die komplette Größe der verwendeten Speicherkarte nutzbar gemacht. Zusätzlich muss der Anwender mitteilen, welche Auflösung das von ihm verwendete Display besitzt: Der FIRE-Taster setzt die Auflösung auf 800x480 Pixel, der ON-OFF-Taster auf 1024x600 Pixel. Ebenso wird die Art des verbauten Funkmoduls abgefragt: Der FIRE-Taster setzt das RFM69HCW, der ON-OFF-Taster das RFM69CW.

Anschließend erfolgt ein automatischer Neustart des Systems, der im Auswahlbildschirm mündet, ob man das Programm starten (FIRE-Button) oder das System herunterfahren (ON-OFF-Button) möchte. Ein gleichzeitiges Drücken beider Taster beendet das Startup-Skript und bringt den Nutzer zur „Bash“, wie die Kommandozeile unter Linux genannt wird.



Abbildung 4.1: Aufgebautes System *raspEasyFire*

## 4.2 Konfiguration und Zugriff

Die meisten Konfigurationseinstellungen sind mittlerweile im Programm via Touchscreen verfügbar, dennoch kann es nötig oder gewünscht sein, unmittelbar auf das Dateisystem oder die Ordnerstruktur des Raspberry Pi zuzugreifen.

Die einfachste Möglichkeit ist der Anschluss einer USB-Tastatur an den Raspberry Pi und Beenden des Startup-Skripts zur Kommandozeile. Alternativ kann auch per SSH über den stationären Rechner auf *raspEasyFire* zugegriffen werden.

Hierzu muss sichergestellt sein, dass sich *raspEasyFire* mit dem Drahtlosnetzwerk verbunden hat, um die lokale IP-Adresse (beispielsweise: 192.168.0.5) ermitteln zu können. Als SSH-Client kann das bereits als Terminalprogramm bekannte **Putty** verwendet werden, welches man mit folgenden Parametern startet:

```
putty.exe -ssh pi@192.168.0.5
```

Nun muss man beim ersten Verbindungsaufbau den SSH-Schlüssel bestätigen und wird dann nach dem Passwort gefragt. Das Standardpasswort für den Nutzer „pi“ lautet *raspberry*. Ab diesem Punkt ist das Vorgehen mit direkt angeschlossener Tastatur oder SSH-Zugang identisch.

## 4.3 Anpassung und Umbenennung von AES-Schlüsseln

Ist der Zugriff eingerichtet, wird eine der ersten Tätigkeiten die Einstellung des passenden AES-Schlüssels sein, damit *raspEasyFire* und die Zündboxen von *El Fueradoro* dieselbe Sprache sprechen.

Ausgangspunkt auf der Bash ist das Home-Verzeichnis des Users „pi“, von dort kann man mit der Eingabe

```
cd raspEasyFire3/config
```

ins Verzeichnis der Konfigurationsdateien wechseln. Auch wenn an dieser Stelle keine ausführliche Linux-Einführung gegeben werden soll, sei kurz darauf hingewiesen, dass Linux im Gegensatz zu Windows bei Datei- und Verzeichnisnamen „case sensitive“ ist, also genau auf Groß- und Kleinschreibung achtet.

In diesem Verzeichnis befinden sich von Haus aus drei Dateien, namentlich

**raspEasyFire.conf** Die Konfigurationsdatei mit grundlegenden Programmeinstellungen. Mit einer Raute (#) beginnende Zeilen stellen dabei Kommentare dar, welche die Bedeutung der nachfolgenden Einstellungen demonstrieren, Einstellungen selbst werden in der Form „Variable: Wert“ angegeben. Typischerweise sollte es nicht notwendig sein, hier Einstellungen zu ändern, der derzeit ausgewählte Zündplan wird beispielsweise immer automatisch durch das eigentliche Programm aktualisiert, andere Einstellungen können in der *raspEasyFire*-Oberfläche geändert werden.

**aesvalues\_default.txt** Diese Datei enthält den Standard-AES-Schlüssel, der zur Kommunikation mit den Zündboxen verwendet wird. Die Werte heißen „value\_01, ..., value\_16“ und müssen im Hexadezimalformat, d.h. mit vorangestelltem „0x“ angegeben werden, der Wertebereich für jeden einzelnen „value“ geht von „0x00“ bis „0xFF“.

**aesvalues.txt** Diese Datei wird enthält den derzeit im Programm verwendeten AES-Schlüssel. Im Auslieferungszustand ist sie mit *aesvalues\_default.txt* identisch.

Um einen eigenen AES-Schlüssel, welcher beispielsweise den Namen „mykey“ tragen könnte, anzulegen, empfiehlt es sich, die Datei *aesvalues\_default.txt* zunächst mit dem Kopierbefehl *cp* zu duplizieren:

```
sudo cp aesvalues_default.txt aesvalues_mykey.txt
```

Das Voranstellen von *sudo* führt dazu, dass der Befehl mit Administratorrechten ausgeführt wird, so dass man sich keine Gedanken um eingeschränkte Schreibrechte machen muss.

Zum Anpassen des Dateiinhalts steht der simple Texteditor „nano“ zur Verfügung, per

```
sudo nano aesvalues_mykey.txt
```

startet man die Bearbeitung.

In „nano“ werden dann die entsprechenden Einstellungen vorgenommen, beendet wird das Programm durch die Tastenkombination „STRG + X“, anschließend bestätigt man die Nachfrage, ob die Änderungen gespeichert werden sollen.

Um die Daten aus *aesvalues\_mykey.txt* zu verwenden, kann man sie entweder später im Programm per Touchscreen auswählen oder direkt via

```
sudo cp aesvalues_mykey.txt aesvalues.txt
```

zur neuen Standardeinstellung machen.

## 4.4 Passwort ändern

Um den Zugriff für unbefugte Personen auf den Raspberry Pi zu verhindern, sollte unbedingt das Passwort des Users „pi“ geändert werden. Möglich ist dies über ein Menü, welches man aus der Bash über

```
sudo raspi-config
```

erreicht.

## 4.5 Zündpläne und Musikdateien übertragen

*raspEasyFire* verarbeitet unmittelbar in *Pyro Ignition Control* erzeigte zpl-Dateien, so dass die Shows bequem am eigenen Rechner konzipiert werden können. Für die Übertragung der Dateien existieren grundsätzlich zwei Möglichkeiten, entweder über einen USB-Stick oder via Netzwerk über das SFTP-Protokoll. Bei Musikdateien muss es sich um mp3-Dateien handeln, welche auch die Endung „.mp3“ besitzen, Zündpläne müssen auf „.zpl“ enden und Dateien, die zusammen gehören, müssen denselben Namen tragen: Eine Show „Feuerwerk.zpl“ verlangt demnach nach einer Musikdatei „Feuerwerk.mp3“.

### 4.5.1 USB-Stick

Um Zündpläne und Musikdateien via USB-Stick zu übertragen, müssen die Dateien im Hauptverzeichnis eines USB-Sticks abgelegt werden und dieser Stick vor dem Hochfahren des Systems mit dem Raspberry Pi verbunden werden. Der Stick wird dann automatisch erkannt und die zpl- bzw. mp3-Dateien in die Verzeichnisse `/home/pi/raspEasyFire3/showfiles/zpl` bzw. `/home/pi/raspEasyFire3/showfiles/mp3` kopiert. Bereits dort existierende Dateien gleichen Namens werden ohne Nachfrage überschrieben.

### 4.5.2 SFTP

Unmittelbar über Netzwerk können Dateien durch ein Programm wie den „FileZilla FTP Client“ verschoben werden. Dort muss dann im Servermanager eine SFTP-Verbindung erstellt werden, wobei der Server der internen IP-Adresse von *raspEasyFire* entspricht und die Zugangsdaten mit Nutzernamen „pi“ und dem entsprechend gewählten Passwort anzugeben sind. Auch hier wird bei erstmaliger Verbindung typischerweise eine Bestätigung gefordert, anschließend kann man durch das Dateisystem navigieren und die Show- und Musikdateien in den entsprechenden Verzeichnissen `/home`/



Abbildung 4.2: Startbildschirm nach dem Aufruf von *raspEasyFire*

`pi/raspEasyFire3/showfiles/zpl` bzw. `/home/pi/raspEasyFire3/showfiles/mp3` ablegen.

## 4.6 Die Benutzeroberfläche

Durch Drücken des FIRE-Buttons nach dem Hochfahren, startet die Benutzeroberfläche von *raspEasyFire*, sie ist in Abbildung 4.2 gezeigt.

Im linken Teil des Bildschirms werden einige Informationen wie der aktuelle Zündplan oder die gemessene Batteriespannung angezeigt, am rechten Rand des Bildschirms sind die Schaltflächen zur Bedienung via Touchscreen zu sehen:

1. Modusauswahl

**Auto-Mode** Vollautomatisches Zünden nach Zeitplan

**Semi-Auto-Mode** Manuelle Zündung per Knopfdruck mit vorgegebener Reihenfolge

**Manual Mode** Vollständig manuelle Zündung

2. Musikbegleitung an/aus

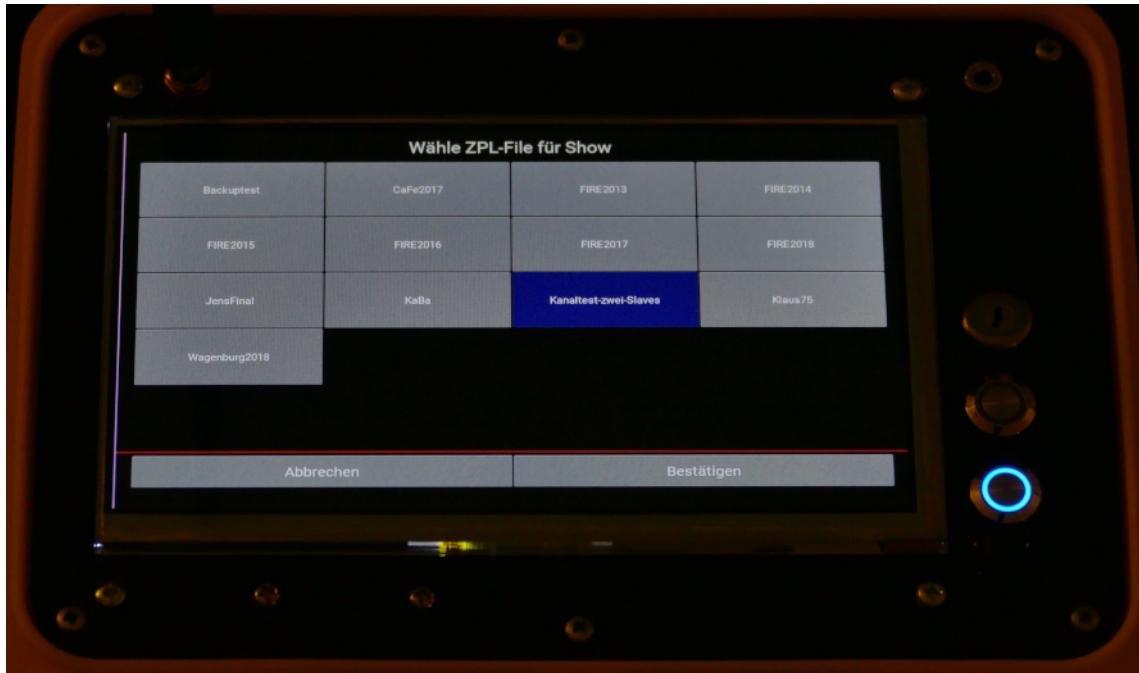


Abbildung 4.3: Auswahl des Zündplans

3. Zündplanauswahl

4. Slavesuche

5. Weitere Optionen

6. Soundcheck

#### 4.6.1 Zündplanauswahl

Anwählen des Punktes „ZPL wählen“ führt zum in Abbildung 4.3 gezeigten Bildschirm. Hier kann aus den im Programmunterverzeichnis „showfiles/zpl“ vorhandenen Zündplänen gewählt werden. Die Auswahl ist abschließend zu bestätigen, um den gewählten Zündplan zu laden.

Analog verhält es sich im manuellen Modus mit Musik mit der Auswahl der MP3-Datei, sofern die Musikbegleitung aktiviert ist.

Die Auswahl von Zündplan bzw. Musikstück wird jeweils gespeichert und ist dann auch beim nächsten Programmstart aktiv.

#### 4.6.2 Slavesuche

Mit einem Druck auf die Schaltfläche „Slaves suchen“ wird eine Suche nach verfügbaren Zündboxen gestartet. Diese dauert etwa sieben Sekunden, anschließend werden die Parameter der in Reichweite befindlichen Boxen wie in Abbildung 4.4 zu sehen auf dem Startbildschirm angezeigt.

Die Anzeige der gefundenen Boxen erfolgt farblich kodiert. Wird eine Box gefunden, deren Slave-ID im aktuell ausgewählten Zündplan Verwendung findet, so werden ihre Daten in weißer Schrift ausgegeben, Boxen mit nicht verwendeter Slave-ID werden in einem Goldton gelistet. Dies dient als Hinweis an den User.

In Klammern hinter „Gefundene Slaves“ taucht zudem die Gesamtanzahl gefundener Zündboxen auf. Werden weniger Boxen als erwartet gefunden, sollten –sofern mehrmaliges erneutes Suchen keinen Erfolg gebracht hat – folgende Ursachen und Lösungsversuche in Erwägung gezogen werden:

1. Box nicht eingeschaltet oder Batterie leer ⇒ Einschalten, Batterie laden/tauschen
2. Unterschiedliche AES-Schlüssel ⇒ AES-Schlüssel abgleichen
3. Fehlende Funkverbindung durch Hindernisse, Abschattung, schlecht angebundene Antenne ⇒ Boxposition anpassen für direkte Sichtlinie, Schraubverbindung der Antenne fest (aber nicht zu fest) anziehen, Abstand zwischen Box und *raspEasyFire* verringern
4. Datenkollision durch gleiche Unique-IDs ⇒ Zuteilung einer anderen Unique-ID

#### 4.6.3 Weitere Optionen

##### Einstellungen anpassen

Unter diesem Punkt können einige Grundeinstellungen des Programms verändert werden:

**Showmodus beim Start** Welche Art der Show soll beim Programmstart standardmäßig eingestellt sein? Diese Einstellung wird erst nach Neustart von *raspEasyFire* aktiv und hat keine Auswirkung auf die aktuelle Einstellung im Programm.

**Ton standardmäßig aus?** Soll die Standardeinstellung „Musik an“ oder „Musik aus“ lauten? Auch diese Einstellung wird erst nach Neustart von *raspEasyFire* aktiv und hat keine Auswirkung auf die aktuelle Einstellung im Programm.

**Eingebautes Funkmodul** Wird mit dem Funkmodul RFM69CW gearbeitet oder mit der Variante RFM69HCW mit höherer Sendeleistung? Eine nicht zur tatsächlich verbauten Hardware passende Einstellung führt dazu, dass die Funkverbindung mit den Zündboxen schon bei geringen



Abbildung 4.4: Anzeige der gefundenen Boxen

Distanzen nicht aufgebaut werden kann.

**Herunterfahren beim Beenden** Soll nach dem Beenden von *raspEasyFire* das System komplett heruntergefahren oder in den Commandprompt gesprungen werden?

**Fehlende Slaves ignorieren** Soll beim Überprüfen der automatischen oder semiautomatischen Show darauf bestanden werden, dass alle im Zündplan genannten Slaves (außer Slave 99) auch tatsächlich mindestens einmal vorhanden und alle Slaves scharf geschaltet sind oder nicht? Für den manuellen Modus hat diese Einstellung keine Bedeutung.

**Soundcheck mit Showmusik** Soll die aktuell ausgewählte Musik für den Soundcheck verwendet werden? Bei „Nein“ wird das Standard-Teststück verwendet.

**Lautstärke** Lautstärkeeinstellung der Soundkarte für die Wiedergabe der Musik in Prozent.

**Maximale Anzahl Slaves** Wie viele Slaves sind zulässig? Dieser Wert ist gleichzeitig auch der Maximalwert für Unique-ID und Slave-ID einer einzelnen Box. Je größer dieser Wert gewählt wird, umso länger dauert auch die Slavesuche, weil das Programm jeder Unique-ID einen Zeitschlitz fester lange zuweist, um auf eine Identifizierungsanfrage zu antworten.

Diese Einstellungen können durch „Bestätigen“ dauerhaft übernommen oder durch „Abbrechen“ wieder verworfen werden.

## Updates durchführen

Bei bestehender Internetverbindung kann hier die Software von *raspEasyFire* auf den letzten Stand gebracht werden. Ein Neustart ist nötig, um die Änderungen wirksam zu machen, der ausgewählte Zündplan wird auf die Standardeinstellung zurückgesetzt, AES-Key und Funkmoduleinstellung bleiben nach einem Update erhalten.

## Verwendeten AES-Key wählen

In diesem Fenster kann gewählt werden, mit welcher der im Unterverzeichnis *config* vorhandenen AES-Schlüssel-Dateien, welche einen Dateinamen der Form *aesvalues\_NAME.txt* tragen müssen, gearbeitet werden soll. Die jeweils aktuell gewählte Datei ist blau markiert.

Sofern keins der Auswahlfelder markiert ist, entspricht der Inhalt der *aesvalues.txt* keiner der vorhandenen Dateien des Namensschemas *aesvalues\_NAME.txt*. Dies sollte man durch Auswahl eines Feldes und anschließendes Bestätigen beheben, um wieder einen definierten Zustand zu erreichen.

## AES-Key mit Box abgleichen

Wird eine eingeschaltete Zündbox über USB-Kabel (v3) bzw. einen RS232-USB-Konverter (v1/v2) an den Raspberry Pi angeschlossen, kann unter diesem Punkt ihr AES-Schlüssel ausgelesen werden. Unterscheidet er sich vom aktuell in *raspEasyFire* eingestellten Schlüssel, kann der Nutzer entscheiden, wessen Schlüssel auf das andere Device übertragen werden soll.

## El Fueradoro flashen

In diesem Untermenü ist es möglich, die aktuellen Bootloader- und Firmwaredateien für *El Fueradoro* herunterzuladen sowie über einen geeigneten Programmer den Bootloader auf ein El-Fueradoro-Device zu schreiben und bei vorhandenem Bootloader die Firmware zu übertragen. Zu diesem Zweck muss das Device von *El Fueradoro* per Programmer (Bootloader) oder USB-Kabel (Firmware v3) bzw. USB-RS232-Adapter (Firmware früherer Generationen) mit *raspEasyFire* verbunden werden.

## Slave ID ändern

Zündboxen können neue IDs zugewiesen werden, den Bildschirm dazu zeigt Abbildung 4.5. Hierzu müssen per „+“ und „-“ die alten und neuen IDs eingestellt werden, durch Druck auf „Bestätigen“ wird der Befehl zur Änderung der IDs dann abgeschickt. Es ist zu beachten, dass diese Änderung nur wirksam werden kann, wenn die entsprechende Zündbox nicht scharf geschaltet ist.

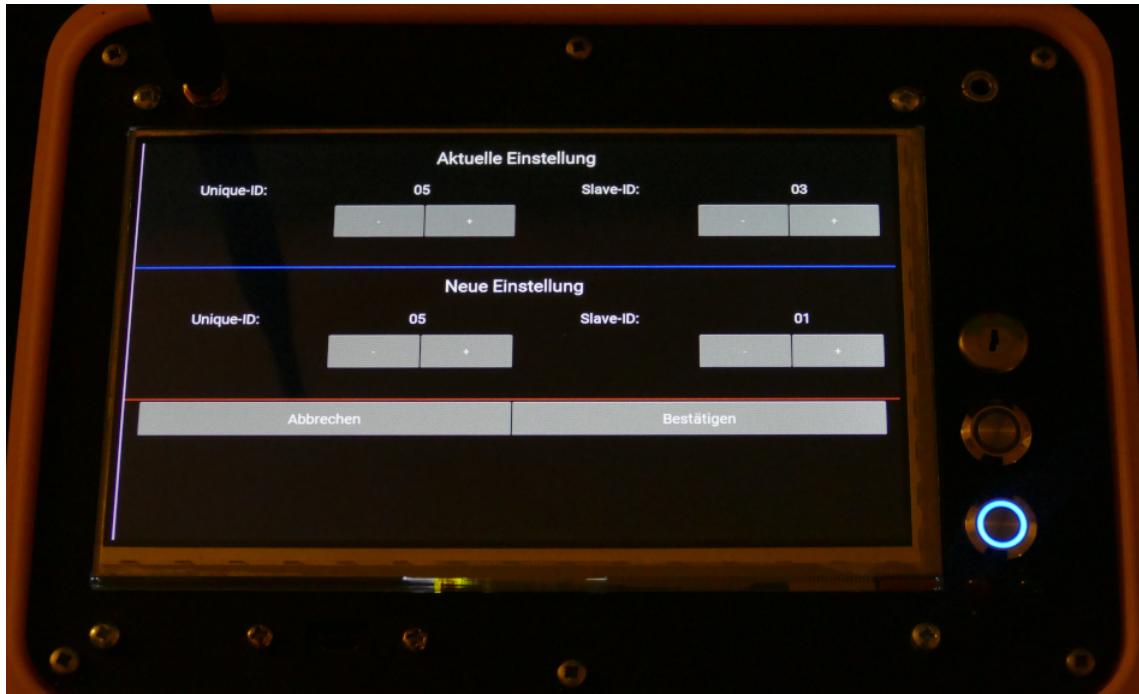


Abbildung 4.5: Ändern der IDs

### Fuoco Step flashen

In diesem Untermenü ist es möglich, die aktuellen Bootloader- und Firmwaredateien für den Stepper *Fuoco Step* herunterzuladen sowie über einen geeigneten Programmer den Bootloader auf den *Fuoco Step* zu schreiben und bei vorhandenem Bootloader die Firmware zu übertragen. Zu diesem Zweck muss der Stepper per Programmer (Bootloader) oder USB-Kabel (Firmware) mit *raspEasyFire* verbunden werden.

### Fuoco Step konfigurieren

Über diesen Menüpunkt können Stepsequenzen im *Fuoco Step* programmiert werden. Auch hierfür ist eine Kabelverbindung nötig, da *Fuoco Step* nicht in der Lage ist, Funksignale zu senden oder zu empfangen.

#### 4.6.4 Dateien löschen

Hier können zpl- und mp3-Dateien gelöscht werden. Nicht löschbar sind Dateien, welche aktuell für die Show oder den Soundcheck ausgewählt sind.



Abbildung 4.6: Widerstandsmessung

#### 4.6.5 Soundcheck

Durch einen Druck auf den Button „Soundcheck“ wird die in der Konfigurationsdatei eingestellte Soundcheckdatei wiedergegeben, möglich ist dies nur im nicht scharfgeschalteten Zustand. Beendet wird der Soundcheck durch Drücken des FIRE-Buttons oder Drehen des Schlüsselschalters in die „scharfe“ Position.

#### 4.6.6 Widerstandsmessung

Durch Anklicken der entsprechenden Zeile in Abbildung 4.4 fordert man eine *El Fueradoro*-Zündbox dazu auf, die Widerstände ihrer 16 bzw. 32 Kanäle zu messen und an *raspEasyFire* zu senden. Diese werden dann in tabellarischer Form angezeigt wie in Abbildung 4.6 gezeigt. Am unteren Rand des Bildschirms kann zur jeweils nächsten und vorherigen Box oder zurück zum Startbildschirm gesprungen werden.

Frühere Generationen der Zündboxen (vor v3) können keine Widerstandsmessung durchführen, sie geben daher für alle Kanäle den Wert „n.a.“ (not available) zurück.

#### 4.6.7 Schlüsselwörter in Zündplänen

Um die Funktionalität der zpl-Dateien zu erweitern, die Kompatibilität mit *Pyro Ignition Control* aber zu erhalten, existieren einige Schlüsselwörter und -zahlen, die für *Pyro Ignition Control* keine Bedeutung haben, in *raspEasyFire* aber entsprechend interpretiert werden:

**STEPPER** Findet sich die Zeichenfolge „STEPPER“ in einer Effektbeschreibung, so signalisiert dies *raspEasyFire*, dass der entsprechende Slave an diesem Kanal einen Stepper angeschlossen hat und der Kanal deshalb unter Umständen mehrfach gezündet werden darf, um den nächsten Effekt am Stepper zu triggern.

**BACKUP** Findet sich die Zeichenfolge „BACKUP“ in einer Effektbeschreibung, wird der Effekt nicht in die automatische Zündliste eingefügt, sondern als Schaltfläche auf dem Touchscreen hinterlegt, von wo aus der dann manuell während der Show durch Drücken des FIRE-Buttons und anschließendes Betätigen der Schaltfläche ausgelöst werden kann. Es spielt dabei keine Rolle, ob die Slave-Kanal-Kombination woanders im Feuerwerk verwendet wird oder nicht, man kann also sowohl zur händischen „Nachzündung“ als auch für zusätzliche Pausen- oder Lückenfüller Backups hinterlegen.

**Slave 999** Durch Verwendung der Slave-ID 999 wird *raspEasyFire* mitgeteilt, dass es sich hierbei nicht um einen tatsächlich zu schießenden Cue handelt, sondern nur um etwas, was auf dem Bildschirm angezeigt werden soll, beispielsweise durch einen zeitgesteuerten Stepper gezündete Cues oder Erinnerungen an eine bestimmte Aktion.

#### 4.6.8 Showstart

Gestartet wird die Show im aktuell eingestellten Modus durch Drücken des FIRE-Buttons, es müssen allerdings bestimmte Voraussetzungen erfüllt sein, damit es wirklich losgehen kann.

Damit die Show reibungslos verläuft, kontrolliert *raspEasyFire*, ob Zündpläne und sonstige Gegebenheiten korrekt sind. Folgende Bedingungen müssen dabei für eine automatische Show erfüllt sein:

- Verwendete Slave-IDs müssen im Bereich von 1 bis zum in den Einstellungen festgelegten Maximum liegen, Slave-ID 999 ist zulässig für „Dummy-Cues“, die zwar auf dem Display angezeigt werden, aber keine Zündung auslösen.
- Ansprechbare Kanalnummern liegen im Bereich von 1 bis 32.
- Jede Kombination aus Slave-ID und Kanalnummer darf im Zündplan lediglich einmal vorkommen. Dient der Kanal der Triggerung eines weiteren Geräts, soll also wirklich mehrfach ausgelöst werden, so ist in die Effektbeschreibung bei jedem Aufruf „STEPPER“ (Großschreibung beachten) mit einzutragen, auch das Hinzufügen von „BACKUP“, falls man bei Versagen

der Zündung noch einmal manuell nachzünden möchte, unterbindet die Fehlermeldung.

- Der Abstand zwischen zwei „echten“ Zündungen muss mindestens 100 ms betragen, für Dummy-Cues gilt diese Beschränkung demnach nicht.
- Für jede angesprochene Slave-ID (Ausnahme ist die Dummy-Slave-ID 999) muss mindestens eine Zündbox mit der entsprechenden Slave-ID gefunden worden sein. Zusätzlich wird geprüft, ob der maximale auf dieser Slave-ID angesprochene Kanalwert von den vorhandenen Boxen erreicht werden kann. Um diese Überprüfung abzuschalten, kann über „Weitere Optionen“ und „Einstellungen anpassen“ die Option „Fehlende Slaves ignorieren“ auf „Ja“ gesetzt werden.
- Alle gefundenen Zündboxen, deren IDs im Feuerwerk verwendet werden, müssen scharf geschaltet sein. Auch diese Überprüfung wird übersprungen, wenn die Option „Fehlende Slaves ignorieren“ auf „Ja“ gesetzt ist.
- Ist eine Musikshow vorgesehen, muss eine zum Zündplan passende Musikdatei im korrekten Verzeichnis vorhanden sein. Ebenso muss die in der Konfigurationsdatei benannte Soundcheckdatei existieren.
- Bei einer Musikshow müssen alle Zündzeitpunkte innerhalb der Dauer des Musikstücks liegen, da die Zündzeiten mit dem Zeitstempel des Audioplayers abgeglichen werden. Möchte man Zündungen nach dem Ende des Musikstücks setzen, muss die mp3-Datei durch das Hinzufügen von Stille am Ende verlängert werden.
- *raspEasyFire* muss durch den Schlüsselschalter scharf geschaltet sein.

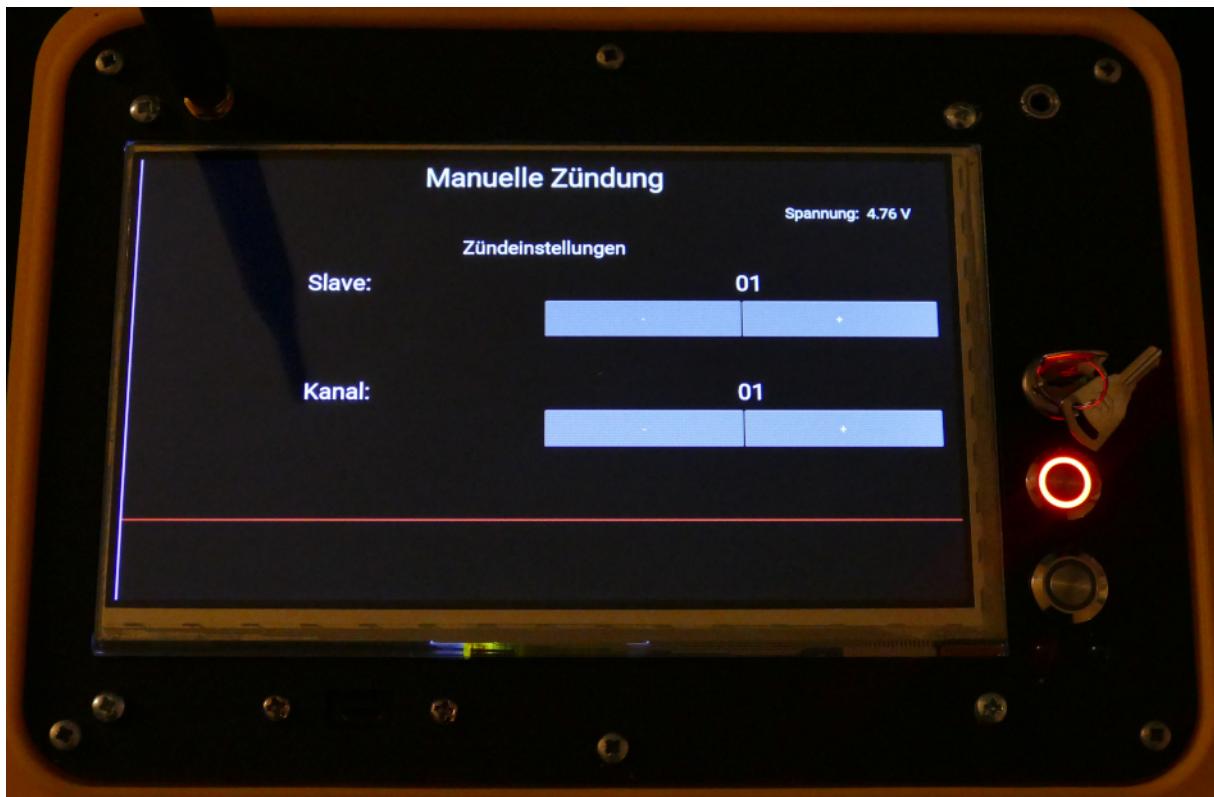
Wird der FIRE-Button gedrückt, ohne dass die nötigen Voraussetzungen erfüllt sind, erscheint ein Info-Popup, welches die zu behebenden Probleme auflistet. Es empfiehlt sich, mit großem Zeitpuffer vor Beginn der Show einen Trockenlauf zu machen, um auf Fehler, die vor Ort nicht mehr allzu leicht zu beheben sind, aufmerksam zu werden. Sind alle Bedingungen erfüllt, beginnt die FIRE-Button-LED nach dem Drehen des Schlüsselschalters in die „scharfe“ Position zu blinken und ein Druck auf den FIRE-Button startet die Show.

Beispiele für die Ansicht im automatischen (mit Backup-Cues) und manuellen Modus sind in Abbildung 4.7a bzw. Abbildung 4.7b gezeigt.

Im automatischen Modus befinden sich in der unteren Bildschirmhälfte zwei Listen: Einerseits die horizontal scrollbare Liste (das automatische Scrollen zum aktuellen Showzeitpunkt kann durch Drücken auf die angezeigte Einstellung deaktiviert oder aktiviert werden) der automatischen Zündungen, darunter die Liste der explizit festgelegten Backup-Cues. Durch Drücken des FIRE-Buttons und anschließendes Betätigen der entsprechenden Schaltfläche kann ein Backup-Cue oder ein Show-Cue erneut ausgelöst werden.



(a) Automatischer Modus



(b) Manueller Modus

Abbildung 4.7: Bildschirme während der Show



# 5 Show

In diesem Kapitel gibt es einige Hinweise zur Planung und Umsetzung eines choreographierten Feuerwerks.

## 5.1 Vorbereitung

### 5.1.1 Beidseitiges Zünden

Gerade in Musikfeuerwerken steht man häufig vor der Herausforderung, ein durch die Dramaturgie des Musikstücks sehr genau abgestecktes Zeitintervall mit Effekten füllen zu müssen. Ist die Laufzeit einer Cakebox länger als dieses Intervall und besitzt der Effekt eine Reservezündschnur, kann die Laufzeit durch eine weitere Zündung an der Reservezündschnur entsprechend verkürzt werden.

Ab dem Zeitpunkt der zweiten Zündung wird die Restlaufzeit der Batterie halbiert, da die Anzündlitze dann von beiden Seiten abbrennt. Bezeichnet man das Zeitintervall zwischen erster und zweiter Zündung mit  $\Delta t$  und die ursprüngliche Laufzeit als  $T_{\text{original}}$ , ergibt sich die neue Laufzeit  $T_{\text{neu}}$ , die zwischen der halben und der ganzen ursprünglichen Laufzeit liegen kann, aus

$$T_{\text{neu}} = \frac{T_{\text{original}} + \Delta t}{2}. \quad (5.1)$$

Durch Umstellen der Formel (5.1) kann der Abstand zwischen den beiden Zündzeitpunkten bei festgelegter neuer Laufzeit bestimmt werden:

$$\Delta t = 2 \cdot T_{\text{neu}} - T_{\text{original}} \quad (5.2)$$

Soll also eine Batterie mit einer Laufzeit von 60 s in 40 s abgebrannt werden, muss die zweite Zündung 20 s nach der ersten ausgelöst werden. Hierbei kommt es natürlich auch zu einer Verdichtung der Effekte.

### 5.1.2 Zündkreisauslegung

An dieser Stelle soll kurz auf die Grenzen und Limits von *El Fueradorno* hinsichtlich der Zündkreise eingegangen werden, die bei der Planung einer Show zu berücksichtigen sind. Die Charakteristika

Strom (A)	Widerstand gesamt ( $\Omega$ )	Widerstand Anzünder + Kabel ( $\Omega$ )
0,6	37,0	34,8
0,75	29,5	27,3
0,9	24,5	22,3
1,0	22,0	19,8
1,2	18,2	16,0
1,5	14,5	12,3
1,75	12,3	10,1
2	10,7	8,5
2,5	8,5	6,3

Tabelle 5.1: Widerstandswerte für Zündkreise bei gegebenem Strom

und Kennwerte der einzelnen Anzündertypen sind Anlage 2 der 1. SprengV zu entnehmen.

*El Fueradoro* ist für die Verwendung mit den Anzündertypen A (Auslösestrom 0,6 A) und U (Auslösestrom 1,3 A) ausgelegt. Der nötige Strom zum Auslösen eines HU-Anzünders (Auslösestrom 25 A) kann nicht geliefert werden. Es ist zu berücksichtigen, dass in der Anlage ein Strombegrenzungswiderstand von  $2,2 \Omega$  verbaut ist, der in Reihe zum angeschlossenen Anzünder-Netzwerk liegt. Mittels Formel (5.3) kann der maximal zulässige Widerstand (inklusive der  $2,2 \Omega$ ) für einen gegebenen mittleren Strom  $I$  ermittelt werden.

$$R_{\max} = \frac{50}{47 \cdot \ln \left( \frac{423 \text{ A}}{423 \text{ A} - 20 \cdot I} \right)} \Omega \quad (5.3)$$

Entsprechend resultiert für einen gegebenen Widerstand der mittlere Strom über eine Zeit von 10 ms aus Gleichung (5.4).

$$I = 21,15 \text{ A} \cdot \left( 1 - e^{-\frac{50 \Omega}{47 \cdot R}} \right) \quad (5.4)$$

Einige Kennwerte für Ströme und die resultierenden maximalen Widerstände sind in Tabelle 5.1 aufgeführt.

Der Kabelwiderstand kann unter Kenntnis von Querschnittsfläche  $A$  und/oder Durchmesser  $d$ , Material (mit spezifischem Widerstand  $\rho$ ) und Kabellänge  $l$  berechnet werden:

$$\begin{aligned} R_{\text{Kabel}} &= \frac{\rho \cdot l}{A} \\ &\text{bzw.} \\ R_{\text{Kabel}} &= \frac{4 \cdot \rho \cdot l}{d^2 \cdot \pi} \end{aligned}$$

Material	$\rho \left( \frac{\Omega \text{ mm}^2}{\text{m}} \right)$
Silber	0,0159
Kupfer	0,0175
Aluminium	0,0265
Eisen	0,15
Stahl	0,2

Tabelle 5.2: Spezifischer Widerstand verschiedener Leitermaterialien

Für einige typische Drahtmaterialien ist der spezifische Widerstand in Tabelle 5.2 dargestellt. Je kleiner der spezifische Widerstand ausfällt, umso größer ist dementsprechend sein Kehrwert, die elektrische Leitfähigkeit.

Bei der Verwendung von Verschleißdraht mit einer Kupferseele mit Durchmesser  $d = 0,5 \text{ mm}$  kann für Überschlagsrechnungen also davon ausgegangen werden, dass je 10 m Kabel-Gesamtweg (also Hin- und Rückleitung addiert) ein zusätzlicher Reihenwiderstand von  $1 \Omega$  im Zündkreis auftritt.

Setzt man pro A-Anzünder einen Widerstand von  $2 \Omega$  und pro U-Anzünder einen Widerstand von  $0,8 \Omega$  an und geht von einem Gesamt-Kabelweg von 10 m aus, sollte die Anlage also im Stande sein, eine Reihe von 10 A-Anzündern, wodurch sich  $R_{\text{ges}} = 23,2 \Omega$  ergibt, noch sicher auszulösen, da der mittlere Strom dann noch mehr als 50 % über dem Auslösestrom eines A-Anzünders liegt. Bei den U-Anzündern ergibt sich unter sonst gleichen Bedingungen, was Stromüberhöhung und Kabellänge angeht, ebenso eine Maximalzahl von 10 U-Anzündern in Reihe.

**Es sei noch einmal deutlich darauf hingewiesen, dass aufgrund der unterschiedlichen Widerstände und Ansprechströme niemals verschiedene Anzündertypen zusammen in einem Zündkreis verwendet werden sollten – und zwar weder in Reihen- noch in Parallelschaltung!**

Eine Beispielrechnung macht dies klar: Würde man beispielsweise je fünf Anzünder jeder Kategorie zu einer Reihenschaltung von 10 Anzündern zusammenschalten, ergäbe sich  $R_{\text{ges}} = 17,2 \Omega$  und damit nach Formel (5.4) ein Strom  $I = 1,27 \text{ A}$ , was bereits unterhalb des spezifizierten Auslösestroms eines U-Anzünders ( $1,3 \text{ A}$ ) liegt.

Hinzu kommt die Problematik, dass selbst bei einem rechnerisch ausreichend hohen Strom dieser in einer „Mischschaltung“ wahrscheinlich nicht über die notwendige Zeitspanne bei den U-Anzündern ankäme. Dies liegt am unterschiedlichen Zündimpuls (Einheit  $\frac{\text{mW s}}{\Omega}$ ), hinter dem sich der bei Schmelzsicherungen als Schmelzintegral bekannte „I-Quadrat-t-Wert“, also das Produkt aus dem Quadrat des Stroms und der Zeit, verbirgt<sup>7</sup>. Fließt ein Strom  $I = 1,3 \text{ A}$  durch die Reihenschaltung eines Anzünders vom Typ A und eines Anzünders vom Typ U, löst der A-Anzünder aufgrund des geringeren notwendigen Zündimpulses zuerst aus. Durch das Auslösen wird in aller Regel der Stromkreis unmittelbar unterbrochen, so dass kein Strom mehr durch den U-Anzünder fließen kann, weshalb die ihn ihm umgesetzte Energie möglicherweise nicht zum Auslösen ausreicht.

<sup>7</sup>  $W = V A$  und  $\Omega = \frac{V}{A} \Rightarrow \frac{W}{\Omega} = A^2$

Die Krux bei einer Parallelschaltung, welche gegenüber der Reihenschaltung ohnehin die schlechtere Alternative darstellt, verschiedener Anzündertypen liegt darin, dass ein Auslösen des Anzünders nicht zwingend gleichbedeutend mit einer Unterbrechung des Stromkreises ist. Bei einer Parallelschaltung unterschiedlicher Widerstände wird zunächst der Ast mit dem geringsten Widerstand auslösen, allerdings kann nicht garantiert werden, dass dieser anschließend auch wirklich einen elektrischen Leerlauf darstellt. Sehr häufig ist auch nach dem Ansprechen noch eine elektrisch leitende Verbindung zwischen den Anzünderdrähten messbar, so dass ein Auslösen der weiteren Äste – gerade im Fall stark unterschiedlicher Widerstände – nicht gesichert ist.

*El Fueradoro* ermöglicht das Auslösen in so kurzen Zeitabständen, dass derartige Drahtseilakte durch eine Aufteilung auf zwei kurz hintereinander erfolgende Zündungen vermieden werden können.

**Dem Anwender wird empfohlen, bei gleichzeitigem Zünden mehrerer Anzünder gleichen Typs eine Reihenschaltung zu verwenden**, da hierdurch die Fehleranfälligkeit und Ausfallwahrscheinlichkeit deutlich verringert wird.

Ein Leerlauf (kein Durchgang eines Anzünders oder schlechte Kabelverbindung) wird bei der Reihenschaltung unmittelbar durch die nicht-leuchtende Kanal-LED bemerkt, bei einem Kurzschluss eines Anzünders fällt lediglich der Effekt dieses einen Anzünders aus. Bei der Parallelschaltung sorgt der Kurzschluss eines Anzünders mit hoher Wahrscheinlichkeit dafür, dass alle parallel geschalteten Anzünder stromlos bleiben, auch wird ein Leerlauf aufgrund des über die anderen Anzünder geschlossenen Stromkreises vermutlich nicht bemerkt.

### 5.1.3 Funkreichweite

Die Frage nach der Funkreichweite, also wie weit die Zündboxen bei *El Fueradoro* vom Transmitter entfernt sein dürfen, um noch sicher auszulösen, kann pauschal nicht beantwortet werden, da dieser Wert sehr vielen Einflussgrößen unterliegt.

Das theoretische Reichweiten-Maximum bei der verwendeten Frequenz von 868 MHz und einer zulässigen Dämpfung von 113 dB (Sendeleistung in dBm minus niedrigste Empfangsleistung in dBm) ergibt sich aus der Friis-Übertragungsgleichung zu mehreren Kilometern. Hierfür müssten die Antennen jedoch mehrere Meter über dem Erdboden angebracht sein, um ideale Wellenausbreitung durch die Luft und Entkopplung vom Boden zu ermöglichen, was in der Feuerwerks-Praxis meist nicht umsetzbar ist, und die Umgebung müsste so störungsfrei sein, dass der Empfänger das Signal auch bei sehr geringen Leistungen noch detektieren kann.

In Tests mit *El Fueradoro* konnte für ein reales Szenario (Platzierung im Gras am Boden) eine Distanz von 250 m zwischen Sender und Empfänger mit einer Reserve von über 20 dB gegenüber der minimalen Empfängerempfindlichkeit überbrückt werden, was für die meisten Feuerwerke ausreichend sein dürfte. Wichtig ist hierbei jedoch eine Sichtverbindung zwischen den Devices. Bricht diese Sichtlinie ab, sinkt die Reichweite, da eine Verbindung über reflektierende und streuende Hindernisse in aller Regel stark verlustbehaftet ist.

Für eine möglichst hohe Reichweite sollte man die Devices daher mit möglichst ungestörter Sichtlinie zueinander und so weit wie möglich vom Boden entfernt platzieren. Die Antennen sollten möglichst parallel zueinander ausgerichtet sein.

In der Systemübersicht, die in Abbildung 3.3 auf Seite 30 zu sehen ist, wird dargestellt, wie stark die anderen Devices vom angeschlossenen Device empfangen werden.

#### 5.1.4 Erstellung und Überprüfung des Zündplans

*Pyro Ignition Control* unterstützt den Ersteller einer Show, indem es sowohl den passenden Zündzeitpunkt aus den Angaben „Effektbeginn“ und „Verzögerung“ berechnet als auch anhand der angegebenen Effektdauer die Überlagerung der Effekte kalkuliert und veranschaulicht. *Pyro Ignition Control* wird den eingegebenen Zündplan perfekt umsetzen, die Krux liegt jedoch darin, dass falsche Eingaben des Erstellers übernommen und nicht hinterfragt werden.

Wenig ist ärgerlicher, als dass eine sorgfältig geplante Show an Flüchtigkeitsfehlern scheitert. Ein falscher Klick bei der Erstellung des Zündplans kann dafür sorgen, dass die Highlightbatterie nicht angesprochen wird und statt buntem Lichterzauber nur schwarze Nacht am Himmel zu sehen ist. Aus diesem Grund existiert das auf *El Fueradoro* zugeschnittene Tool **schemecheck.exe**, welches folgende Dinge im Zündplan erkennen kann und hinweist auf:

- Verwenden einer Slave-ID außerhalb des Bereichs 1 – 30
- Ansprechen eines Kanals außerhalb des Bereichs 1 – 32
- Unterschreiten des minimalen Zündabstands von 100 ms
- Mehrfachzünden derselben Slave-Kanal-Kombination

Darüber hinaus werden auch alle angesprochenen Slave-IDs und die Gesamtzahl der Cues angezeigt.

Zur Überprüfung muss der in *Pyro Ignition Control* erstellte Zündplan als zpl-Datei gespeichert und dem Kommandozeilentool folgendermaßen übergeben werden:

*schemecheck.exe fw.zpl*

## 5.2 Durchführung

Die Durchführung eines Feuerwerks mit allen bürokratischen, logistischen, sicherheitstechnischen und sonstigen Herausforderungen ist Gegenstand eigener Lehrbücher und Vorschriften, weshalb hier nur auf den Einsatz von *El Fueradoro* eingegangen werden soll.

Hierbei gilt ebenso wie in vielen anderen Lebensbereichen der Spruch „Ordnung ist das halbe Leben“. Die Anzünderkabel sollten beschriftet und derart verlegt werden, dass man bei einer eventuellen Fehlersuche schnell fündig wird, zudem sollte ihre Länge so kurz wie möglich aber so lang wie nötig dimensioniert werden, um keine zu hohen Leitungswiderstände zu produzieren, aber auch keine Stolperfallen.

Vor Aufbau der Zündkoffer gilt es, für jede Zündbox eine geeignete Position mit genügendem Abstand zu den Effekten zu finden, um ein Übergreifen entstehender Brände auf den Koffer und die Zündbox zu verhindern. An diesen Positionen werden die Enden der Anzünderkabel für jede einzelne Box gesammelt.

Nun kann der jeweilige Koffer platziert und geöffnet werden, im ersten Schritt wird die Antenne angebracht und fest an der Buchse an der Zündbox verschraubt, ehe man die metallischen Teile der Schraubverbindung mit einem Klebestreifen gegen Berührungen mit freiliegenden Kabeln isoliert. Anschließend kann der Akku verbunden werden. Um böse Überraschungen zu vermeiden, sollte man die Box einschalten und mit dem Transmitter kontrollieren, ob sie an dieser Position erreichbar ist.

Im nächsten Schritt muss, falls noch nicht geschehen, der Schlüsselschalter auf Position „rot“ gedreht und abgezogen werden. Erst dann dürfen die Anzünderkabel angeschlossen werden, zunächst an der schwarzen, dann an der zugehörigen roten Klemme. Sind alle Kabel angeklemmt, kann man die Box einschalten, dabei durch das Blinken der gelben und orangen Status-LEDs sicherstellen, dass sie auf die korrekte Slave-ID programmiert ist, und anhand der Kanal-LEDs überprüfen, ob auf allen angeschlossenen Kanälen ein geschlossener Stromkreis vorliegt. Ist letzteres nicht der Fall, müssen zunächst Kabelverbindungsstellen kontrolliert werden. Erweisen die sich als in Ordnung und sind auch die einzelnen Kabelwege niederohmig, so dass ein Kabelbruch ausgeschlossen werden kann, müssen die Anzünder kontrolliert und eventuell getauscht werden. Zum Schutz der Zündbox vor Wetter und sonstigem Niederschlag ist der Kofferdeckel nach Abschluss der Arbeiten zu schließen, vorher kann die Box zur Batterieschonung erst einmal wieder abgeschaltet werden. Ein Abdecken der Boxen mit Brandschutzmatten kann in Erwägung gezogen werden, die Antenne sollte zwecks Empfang aber immer frei sein und senkrecht nach oben zeigen.

Das Scharfschalten der Boxen darf als allerletzter Schritt erst unmittelbar vor dem Beginn der Show erfolgen. Mit einem Identifikationsbefehl vom Transmitter und anschließender Betrachtung der Empfängerliste kann kontrolliert werden, ob alle Boxen erreichbar und scharf geschaltet sind. Fällt auf, dass eine Box die falsche Slave-ID besitzt, muss sie zunächst wieder entschärft werden, bevor man einen Remote-ID-Wechsel durchführen kann.

Sind diese Arbeiten erledigt und der Transmitter scharf geschaltet, kann das Terminalprogramm/die GUI beendet werden und *Pyro Ignition Control* gestartet werden. Hier den Zündplan öffnen, verbinden und scharfschalten – jetzt kann der Spaß endlich losgehen!

## **Teil II**

# **Dokumentation**



## 6 Schaltpläne & Layouts

In den Abbildungen 6.1, 6.2, 6.3, 6.4., 6.5 und 6.6 sind die Schaltpläne des PC-Transmitters, der verschiedenen Zündbox-Generationen und des Raspberry-Pi-Transmitters nach Funktionseinheiten unterteilt gezeigt, in den Abbildungen 6.7, 6.8, 6.9, 6.10, 6.11 und 6.12 die Layouts.

Erstellt wurden diese mit den Layoutprogrammen **EAGLE** von CadSoft bzw. **KiCAD**, die Originaldateien sowie Gerber-Dateien zur Fertigung sind im Unterordner „Schematics\_and\_Layouts“ des Projekt-Hauptverzeichnisses abgelegt.

Die Layouts der ersten Generation wurden dabei so gestaltet, dass die Platinen nicht zwingend zweiseitig gefertigt werden müssen. Die Anzahl der Leiterbahnen auf der Oberseite wurde minimiert, zudem verlaufen sie nicht unterhalb von Bauelementen und können daher als Drahtbrücken ausgeführt werden. Die nachfolgenden Generationen der Zündboxplatine und die Adapterplatine für den Raspberry Pi wurden hingegen als typische Zwei-Lagen-Platine entworfen.

Die Abmessungen der Platinen finden sich in Abschnitt 11 auf Seite 99.

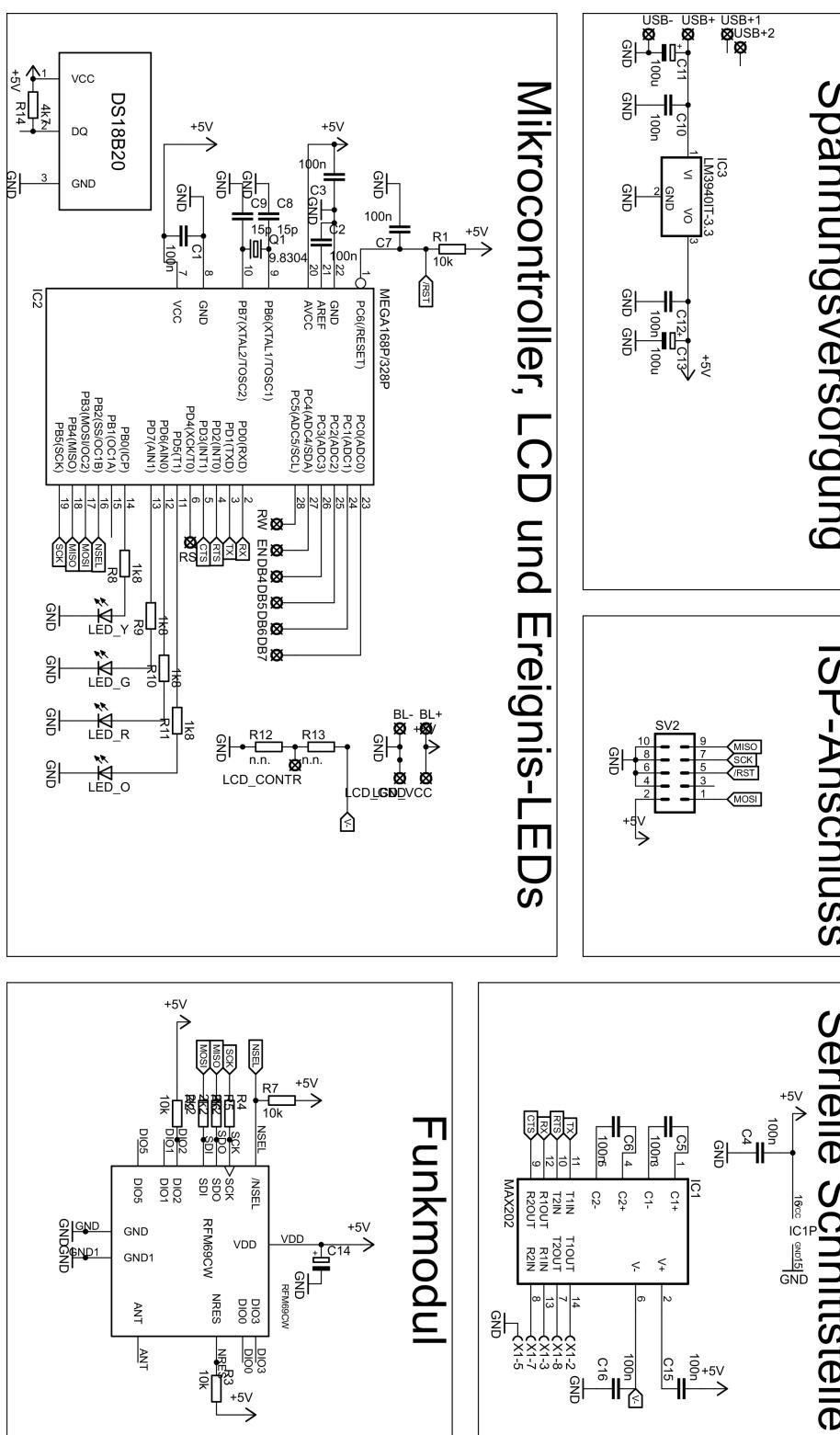


Abbildung 6.1: Schaltplan des Transmitters

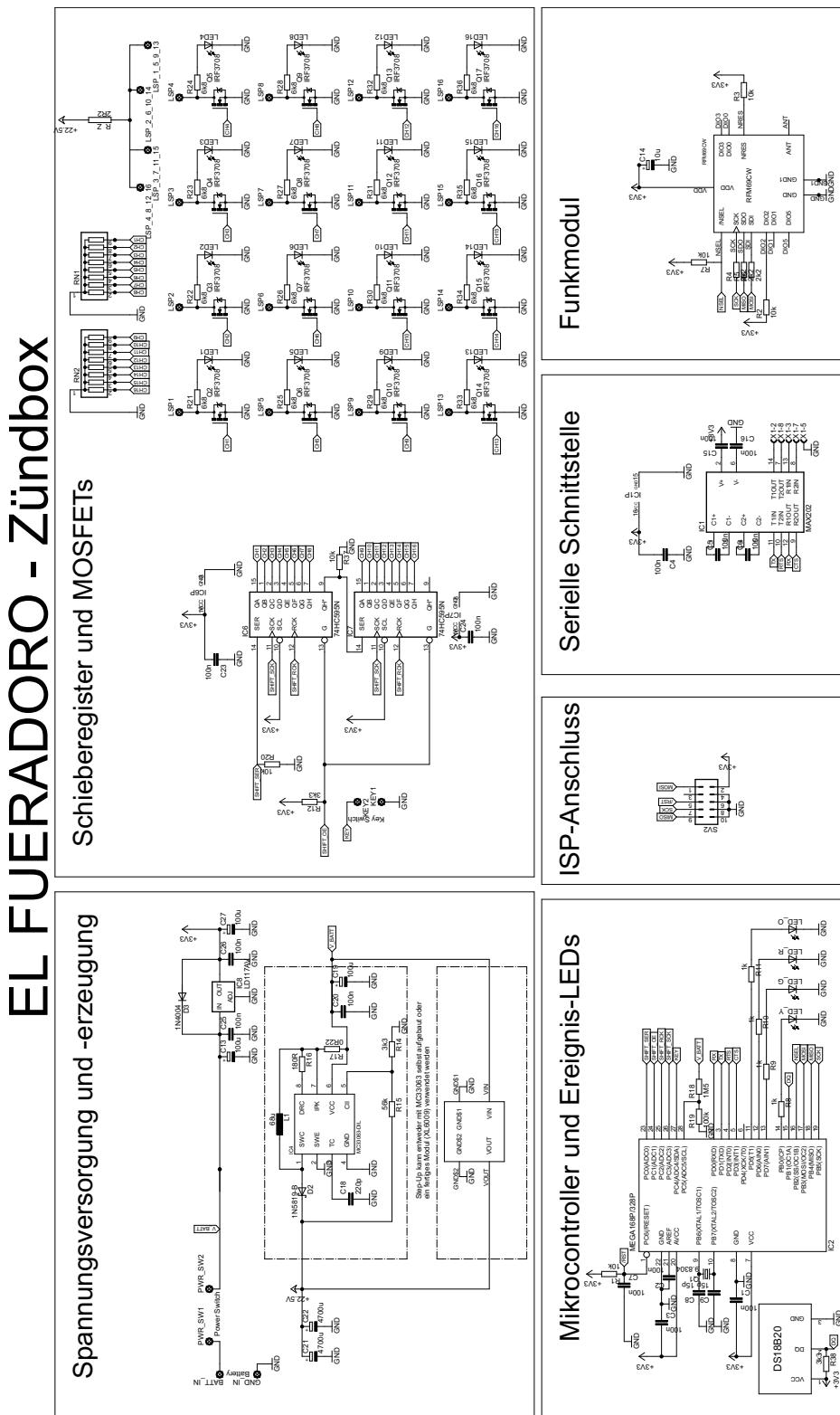


Abbildung 6.2: Schaltplan der Zündbox

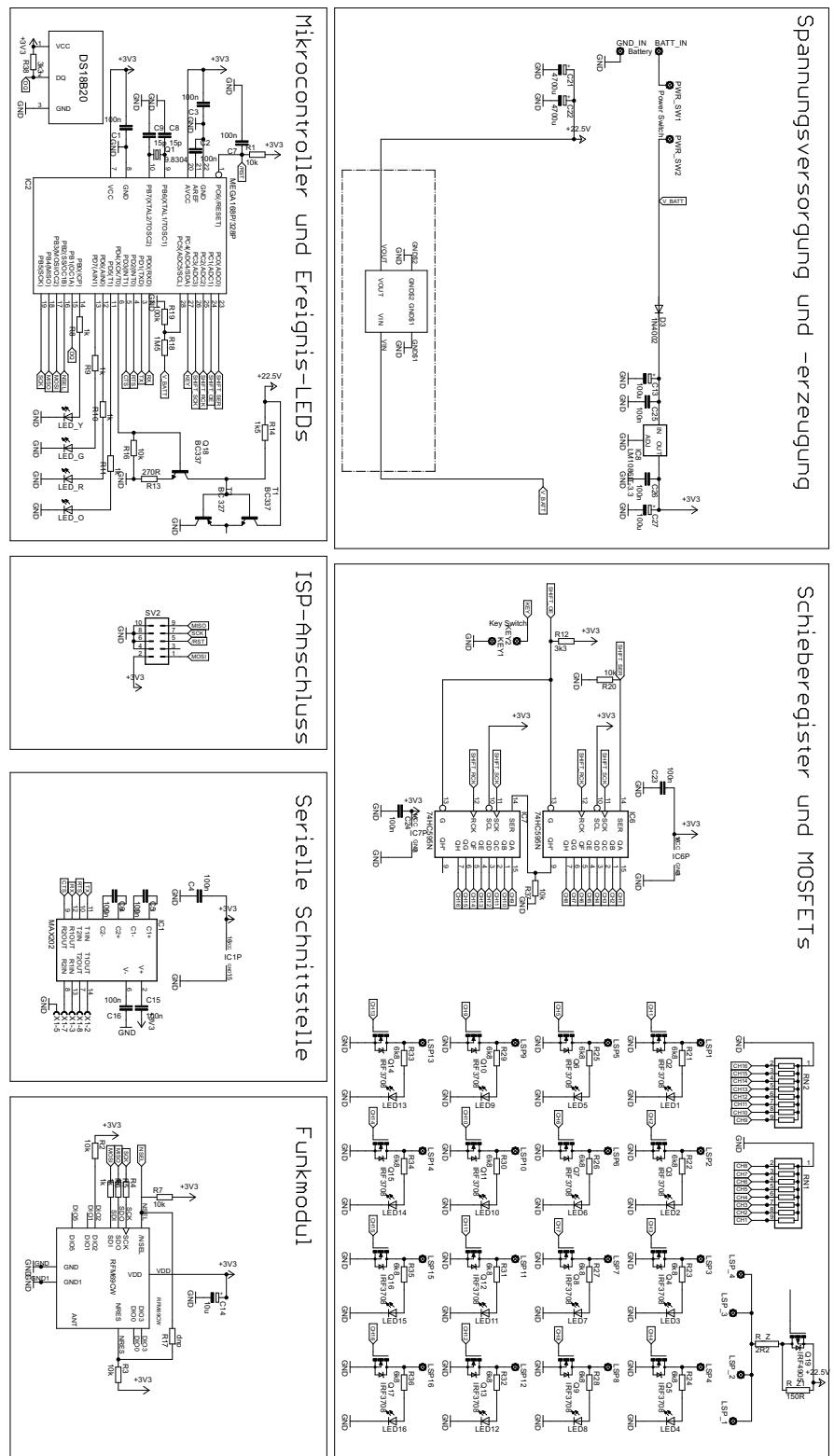


Abbildung 6.3: Schaltplan der Zündbox (2. Generation)

# El Fueradorno Schaltplan Zündbox v3

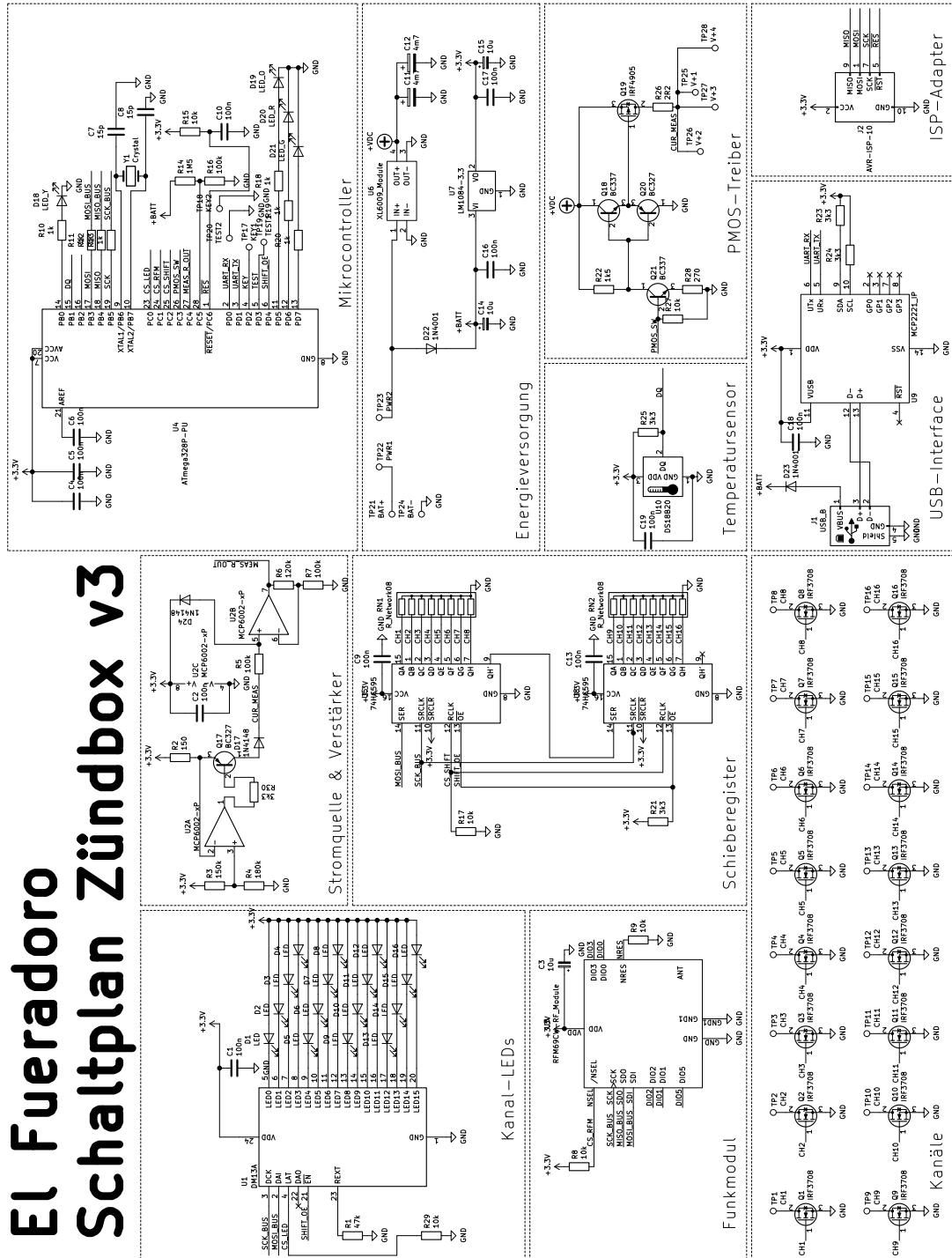


Abbildung 6.4: Schaltplan der Zündbox (3. Generation)

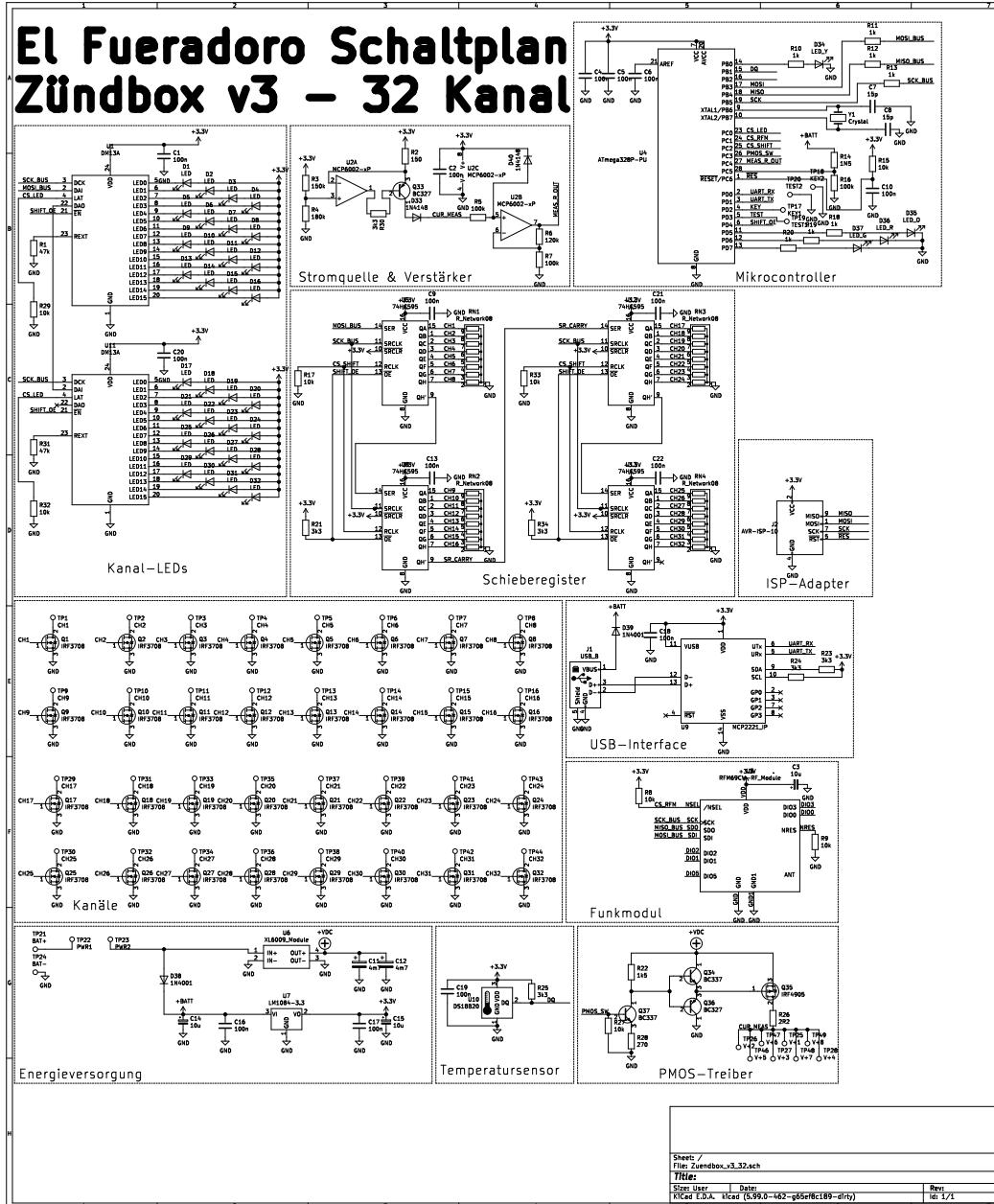
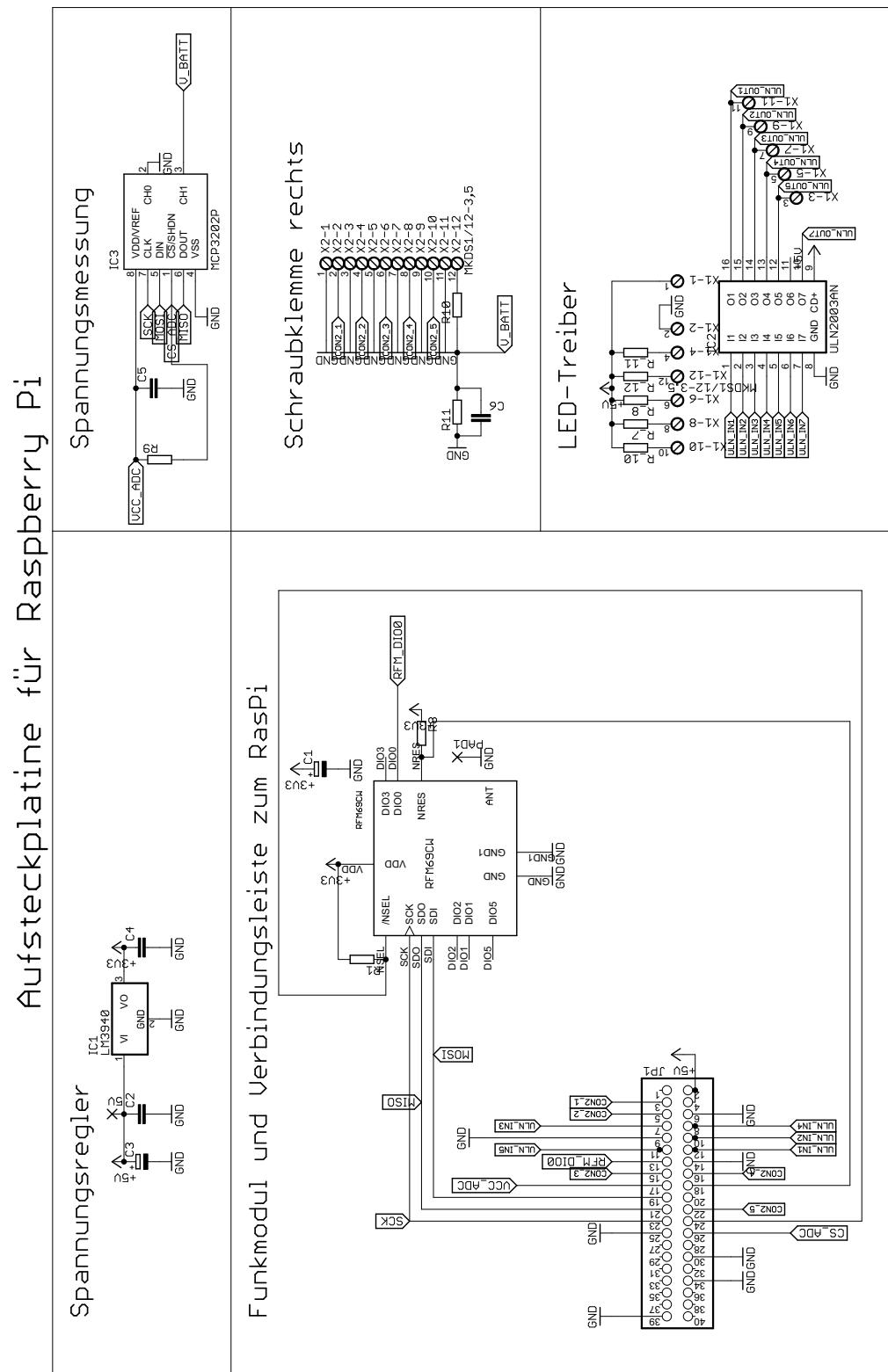


Abbildung 6.5: Schaltplan der Zündbox (3. Generation) mit 32 Kanälen



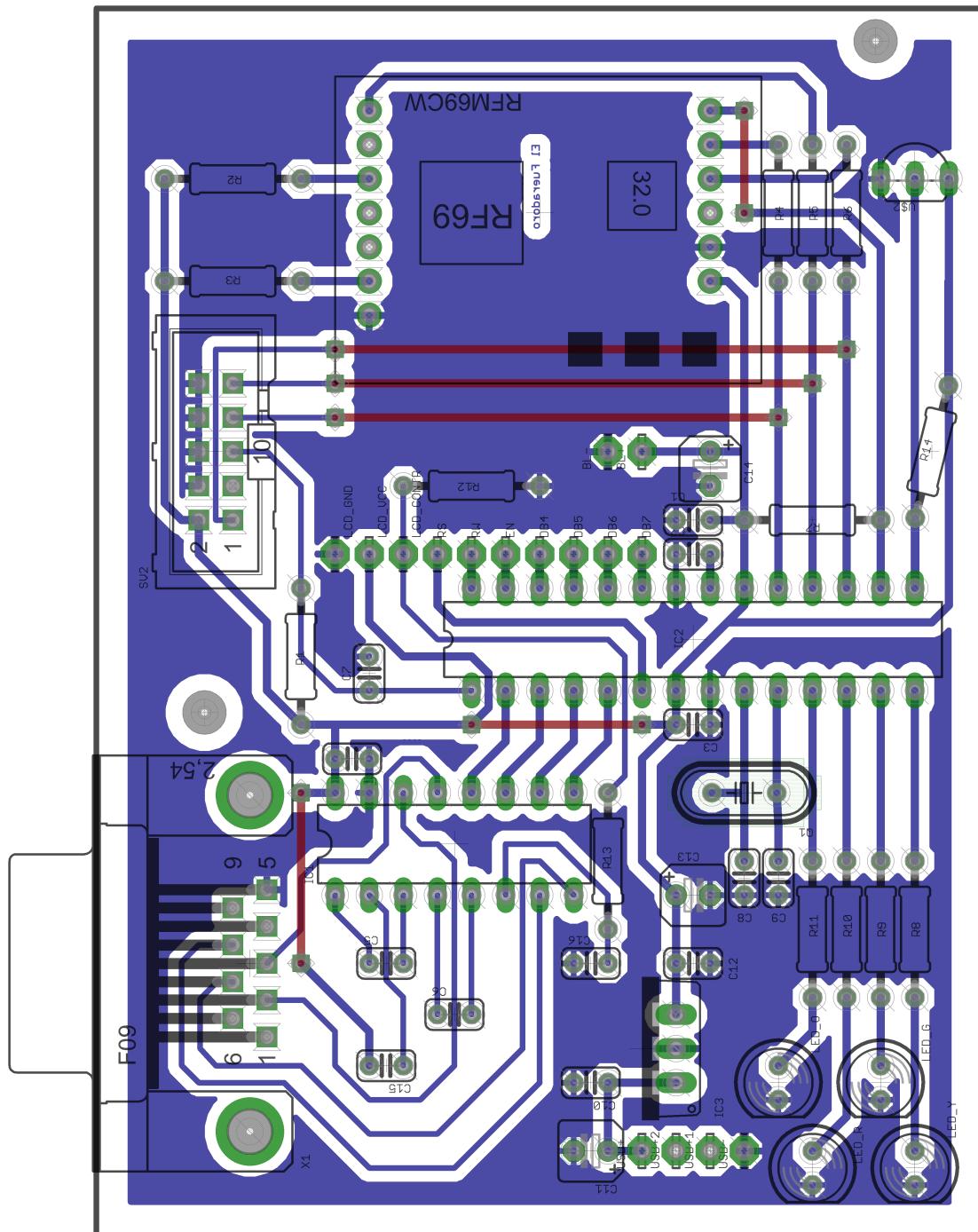


Abbildung 6.7: Layout des Transmitters (keine Originalgröße!)

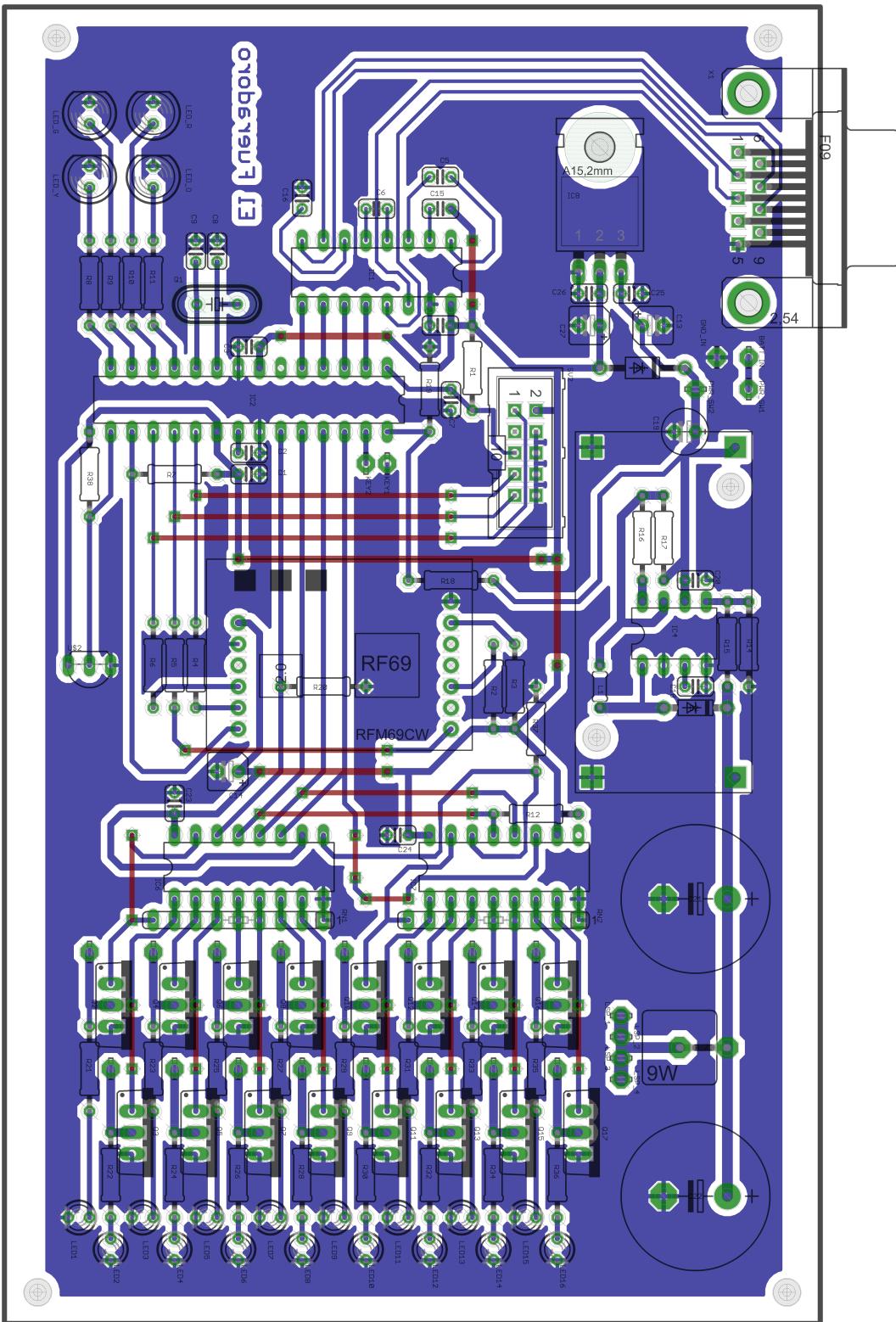


Abbildung 6.8: Layout der Zündbox (keine Originalgröße!)

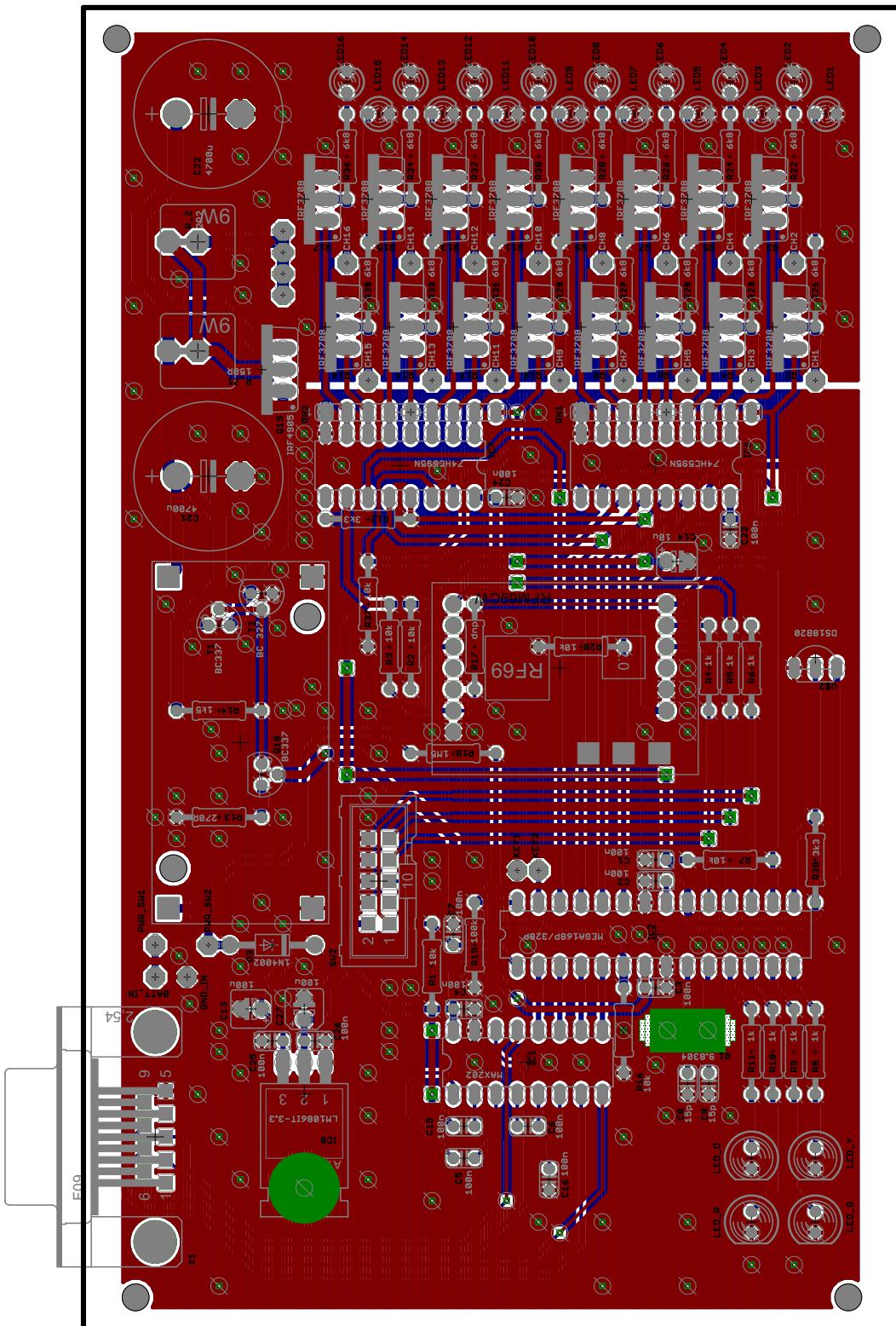


Abbildung 6.9: Layout der Zündbox (2. Generation, keine Originalgröße!)

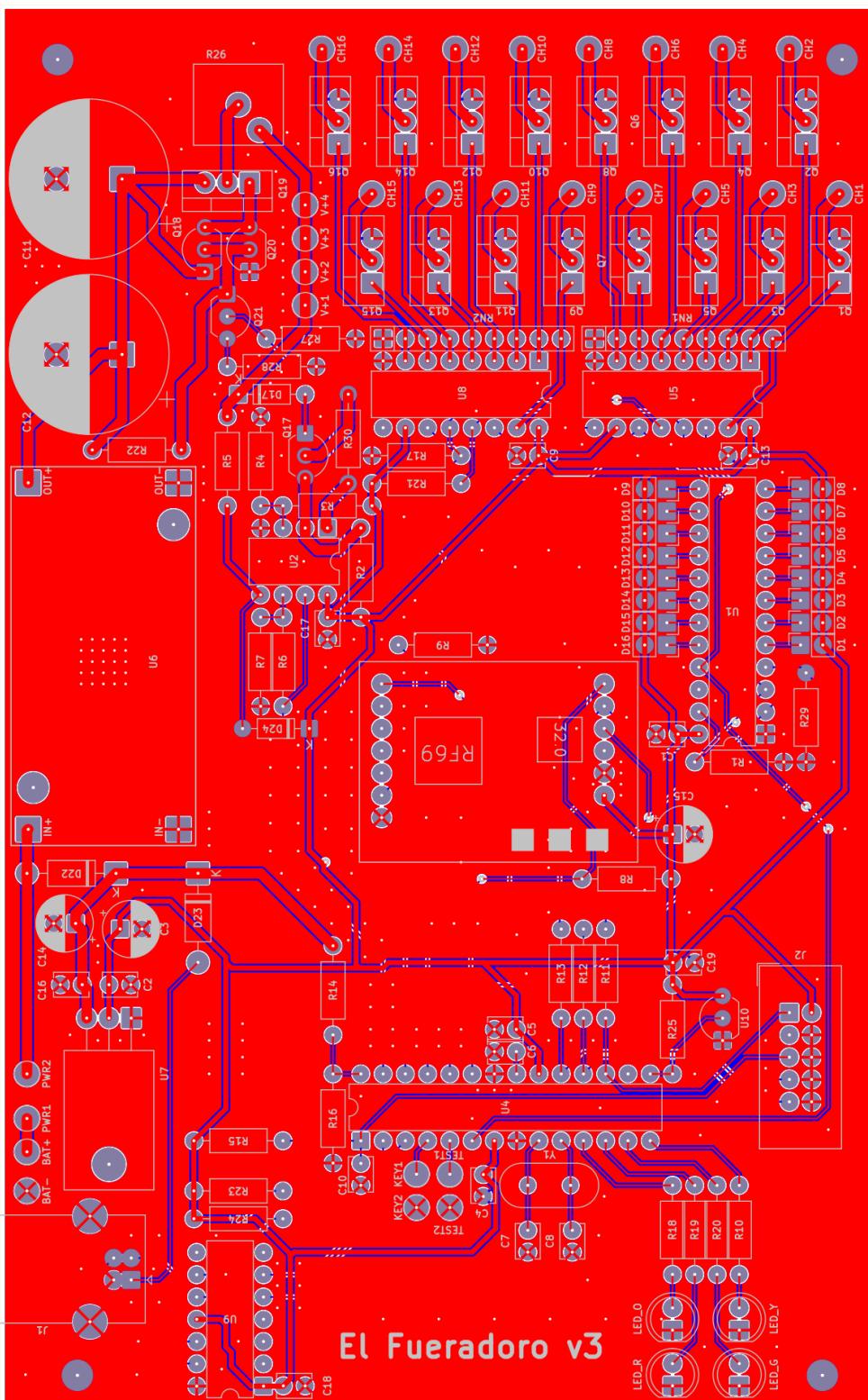


Abbildung 6.10: Layout der Zündbox (3. Generation, keine Originalgröße!)

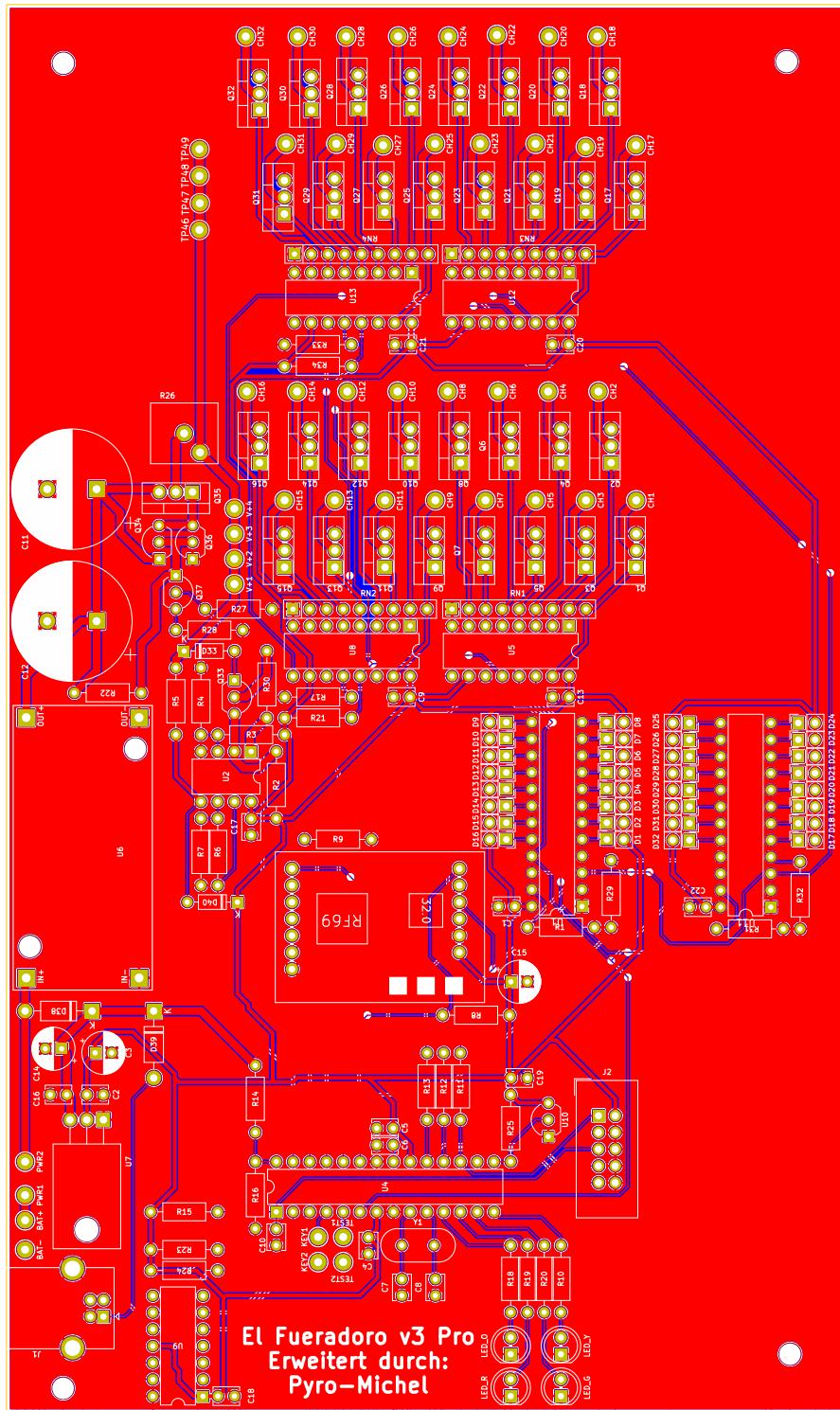


Abbildung 6.11: Layout der Zündbox (3. Generation mit 32 Kanälen, keine Originalgröße!)

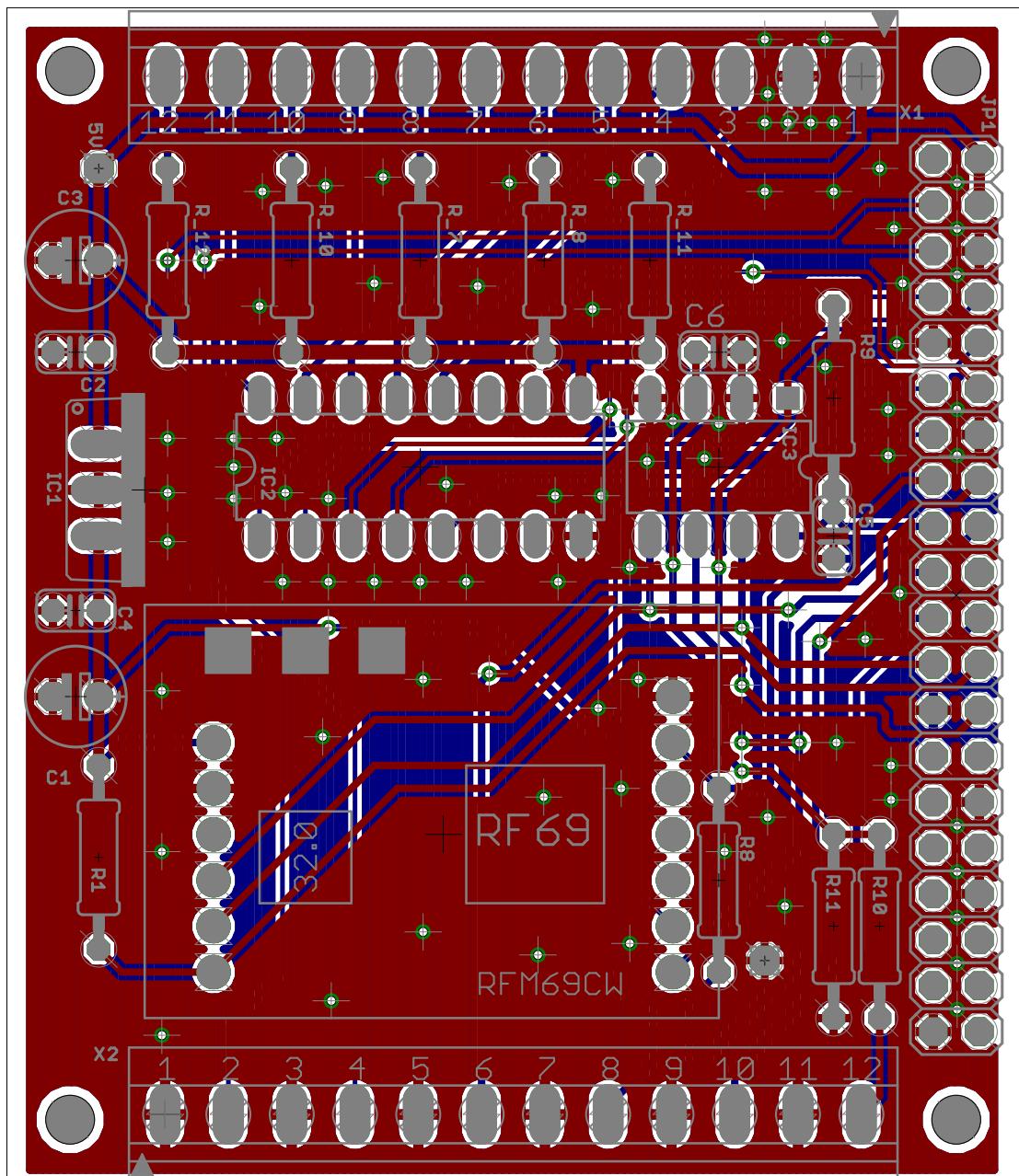


Abbildung 6.12: Layout des Raspberry-Pi-Adapters (keine Originalgröße)



## 7 Datenblätter

- [1] Atmel. *ATmega48A/PA/88A/PA/168A/PA/328/P*. Nov. 2015. URL: [http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p\\_datasheet\\_complete.pdf](http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p_datasheet_complete.pdf).
- [2] HopeRF. *RFM69CW ISM TRANSCEIVER MODULE*. Dez. 2013. URL: <http://www.hoperf.com/upload/rf/RFM69CW-V1.1.pdf>.
- [3] Texas Instruments. *LM1086 1.5-A Low Dropout Positive Regulators*. Apr. 2015. URL: <http://www.ti.com/lit/ds/symlink/lm1086.pdf>.
- [4] Texas Instruments. *LM3940 1-A Low-Dropout Regulator for 5-V to 3.3-V Conversion*. Feb. 2015. URL: <http://www.ti.com/lit/ds/symlink/lm3940.pdf>.
- [5] Texas Instruments. *SNx4HC595 8-Bit Shift Registers With 3-State Output Registers*. Sep. 2015. URL: <http://www.ti.com/lit/ds/symlink/sn74hc595.pdf>.
- [6] Maxim Integrated. *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*. Apr. 2008. URL: <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>.
- [7] International Rectifier. *IRF3708PbF SMPS MOSFET*. Aug. 2007. URL: <http://www.irf.com/product-info/datasheets/data/irf3708pbf.pdf>.
- [8] ON Semiconductor. *MC34063A, MC33063A, SC34063A, SC33063A, NCV33063A – 1.5 A, Step-Up/Down/Inverting Switching Regulators*. Aug. 2010. URL: [http://www.onsemi.com/pub\\_link/Collateral/MC34063A-D.PDF](http://www.onsemi.com/pub_link/Collateral/MC34063A-D.PDF).



## 8 Pinbelegung

## 8.1 Mikrocontroller

Die Pinbelegung des ATmega328p in den verschiedenen Devices ist in Abbildung 8.1 gezeigt.

Besondere Bedeutung kommt bei Transmitter und den ersten beiden Zündboxgenerationen dem Pin rechts oben (LCD-RW bzw. V\_BATT) zu, da er beim Starten der Firmware als Eingang geschaltet wird und aufgrund der dort anliegenden Spannung die sicherheitsrelevante Erkennung, um welchen Devicetyp es sich handelt, vorgenommen wird. Aufgrund eines internen Pullup-Widerstands im LCD werden bei angeschlossenem Transmitter stets 3,3 V an diesem Pin anliegen, bei Zündboxen bewegt sich die heruntergeteilte Spannung der Versorgungsbatterie im Bereich unterhalb von 1,1 V.

Diese Unterscheidung ist wichtig, da der Programmablauf sich bei Transmittern (Unique-ID 0 und Slave-ID 0) anders gestaltet als bei Zündboxen und Controllerpins, wie in Abbildung 8.1 zu sehen, bei Transmittern anders belegt sind und für eine andere Datenrichtung (Eingang/Ausgang) ausgelegt sind als bei Zündboxen. Wie zu erkennen werden Controlleranschlüsse beim Transmitter als Steuerung des Displays verwendet, die bei der Zündbox den Zustand des Schlüsselschalters einlesen oder die Schieberegister zur Zündung der Kanäle ansteuern.

Würde die Software beim Start nicht überprüfen, auf welcher Art Device sie gerade läuft, könnte das Programm bei falscher Konfiguration davon ausgehen, auf einem anderen Device zu laufen. Während die Zündbox-Konfiguration auf einem Transmitter – abgesehen davon, dass das LCD nichts

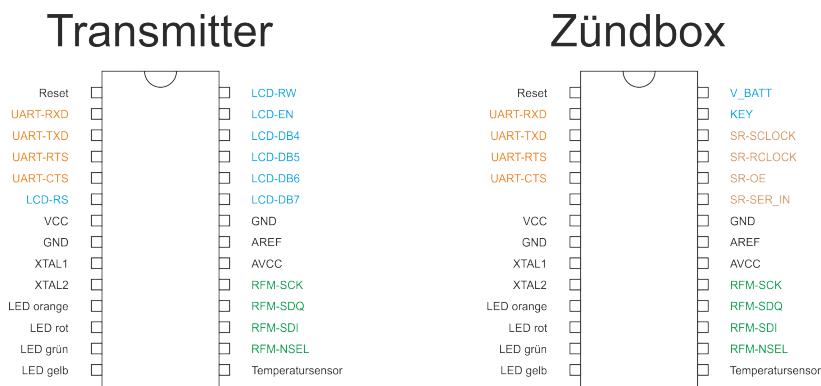


Abbildung 8.1: Pinbelegung des Mikrocontrollers bei Transmitter und Zündboxen

anzeigen würde – keine Probleme hervorriebe, würde es zu Schäden am Controller kommen und zu unerwünschten Zündungen führen, wenn die Transmitter-Konfiguration auf einer Zündbox gestartet werden würde: Im Fall des Schlüsselschalter-Pins käme es zu einem Kurzschluss, wenn die Box scharf geschaltet ist, was den Controller beschädigen kann, die Ansteuerung des Schieberegisters mit LCD-Befehlen würde dazu führen, dass Zündkanäle durchschalten, weil die Ansteuerbefehle des LCD ans Schieberegister weitergeleitet werden.

Aufgrund der einschneidenden Änderungen in der dritten Zündboxgeneration weicht die Pinbelegung stark ab und die Firmware der Zündbox\_v3 ist dementsprechend nicht mit anderen Devices kompatibel!

Für den Betrieb der Devices von *El Fueradoro* müssen nicht zwingend die Standardlayouts verwendet werden, die Firmware kann, sofern die Peripherie mit den „richtigen“ Pins verbunden ist, auch auf anderen Boards wie z. B. Arduino laufen.

## 8.2 Raspberry-Pi-Aufsteckplatine

Die in Abbildung 6.12 gezeigte Platine verfügt über die in den Abbildungen 8.2 und 8.3 gezeigte Pinbelegung. Sie ist vom Formfaktor so konzipiert, dass sie mit einer nach unten gerichteten Buchsenleiste auf die Stifteleiste des Raspberry Pi aufgesteckt und an den vier Ecken über insgesamt 11 mm hohe Abstandshalter mit Gewindeschrauben M2,5 mit dem Einplatinencomputer verschraubt werden kann.

Die an den Seiten der Aufsteckplatine angebrachten Schraubklemmen dienen dem Anschluss der am Gehäuse verbauten Indikatoren (LEDs) und Schalterelemente. Welches Signal an welcher Stelle anzuklemmen ist, illustriert Abbildung 8.3, welche eine Draufsicht auf die Oberseite der Aufsteckplatine zeigt.

Die 40-polige Buchsenleiste zeigt dabei in die Papierebene hinein, die Klemmen an der Seite aus der Papierebene heraus.

Supply	3V3	1	2	5V	Supply
In 1 (FIRE-Bt.)	R1	3	4	5V	Supply
In 2 (ON/OFF-Bt.)	R2	5	6	GND	Supply
Out 3 (Rx-LED)	L3	7	8	L4	Out 4 (Tx-LED)
Supply	GND	9	10	L2	Out 2 (ON/OFF-LED)
Out 5	L5	11	12	L1	Out 1 (Fire-LED)
DIO0 RFM69	DIO0	13	14	GND	Supply
In 3	R3	15	16	R4	In 4
Supply	3V3	17	18	RES	Reset RFM69
SPI Master Out	MOSI	19	20	GND	Supply
SPI Slave Out	MISO	21	22	R5	In 5
SPI Clock	SCLK	23	24	CS0	SPI Select RFM
Supply	GND	25	26	CS1	SPI Select ADC
		27	28		
		29	30	GND	Supply
		31	32		
		33	34	GND	Supply
		35	36		
		37	38		
Supply	GND	39	40		

Abbildung 8.2: Belegung Verbindungsstiftleiste Pi-Aufsteckplatine

	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	
1		3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	
1	U+																				1
2	GND	U <sub>Force</sub>																			2
3	-																				3
4	+	L5 (Pin 11)																			4
5	-																				5
6	+	L4 (Pin 8) TX-LED																			6
7	-																				7
8	+	L3 (Pin 7) RX-LED																			8
9	-																				9
10	+	L2 (Pin 10) ON/OFF-LED																			10
11	-																				11
12	+	L1 (Pin 12) FIRE-LED																			12

Abbildung 8.3: Pinbelegung der Aufsteckplatine von oben gesehen



## 9 Software

Die Steuerungssoftware für *El Fueradoro* ist unter Zuhilfenahme des AVR-GCC in der Programmiersprache C geschrieben. Sie umfasst für die dritte Zündboxgeneration insgesamt 16, ansonsten 15 Sourcefiles (.c) mit zugehörigen Headerdateien (.h), eine Headerdatei zur Generierung von Registeradressen „portmakros.h“ sowie eine globale Headerdatei „global.h“, in welcher alle anderen erfasst sind.

Die Quellcodedateien und ihre Aufgaben sind in Tabelle 9.1 aufgelistet.

Dateiname	Aufgabe(n)
pyro.c	Hauptprogramm, Interruptroutinen und anlagenspezifische Funktionen (Schalterinitialisierung, spezielle LCD-Symbole, ...)
1wire.c	Steuerung des Temperatursensors DS18B20
adc.c	Erfassung der Versorgungsspannung und des Devicetyps mittels Analog-Digital-Converter
addresses.c	Unique- und Slave-ID aus dem Speicher holen, speichern, überprüfen
crcchk.c	Überprüfen der Korrektheit empfangener Zeichenketten
dm13a.c	Steuerung der Kanal-LEDs (nur in der dritten Zündboxgeneration vorhanden)
eeprom.c	Direkter Zugriff auf den EEPROM des Controllers
lcd.c	Steuerung des LCD
leds.c	Kontrolle der vier Status-LEDs
rfm69.c	Funktionen für das Funkmodul RFM69CW
shiftregister.c	Schieberegister-Initialisierung und -Datenübertragung
terminal.c	„GUI“ zur Benutzerinteraktion via Terminalprogramm
timer.c	Funktionen zur Timer-Steuerung
uart.c	Kommunikation über serielle Schnittstelle

Tabelle 9.1: Quellcodedateien und ihre Funktionen



## **Teil III**

# **Aufbauanleitung**



## 10 Materiallisten

In den Tabellen dieses Kapitels sind die benötigten Teile für den Aufbau der verschiedenen Devices von *El Fueradodo* zusammen mit möglichen Bezugsquellen und Preisen (Stand Januar 2015, teils aktualisiert im August 2015) aufgelistet. Bei den Zündboxen wird die zweite Generation empfohlen, die zusätzliche Sicherheit gegenüber der ersten Generation bietet. Es ist jedoch problemlos möglich, beide Generationen gemeinsam in einer Show zu verwenden.

Leider ist es – schon alleine aufgrund der Währungsschwankungen, denen besonders die eBay-Artikel unterworfen sind – ein Ding der Unmöglichkeit, die Liste permanent tagesaktuell zu halten. Tote Links dürfen gerne gemeldet werden.

**Materialliste Transmitter**

Bauteil	Anz.	Beschreibung	Händler	Artikelnr.
Kondensatoren				
C11, C13	2	Elektrolytkondensator, 100uF	Reichelt	RAD 105 100/35
C14	1	Elektrolytkondensator, 10uF	Reichelt	RAD 10/100
C1...7, C10, C12, C15,	11	Keramikkondensator, 100nF	Reichelt	X7R-2,5 100N
C16 C8, C9	2	Keramikkondensator, 15pF	Reichelt	KERKO 15P
Integrierte Schaltungen				
IC2	1	ATMEGA 328P	Reichelt	ATMEGA 328P-PU
IC1	1	MAX202	Reichelt	MAX 202 ECPE
U\$4	1	RFM69CW	Pollin	810 303
U\$2	1	DS18B20	Reichelt	DS 18B20
IC8	1	LM3940-3,3	Reichelt	LM 3940 IT3,3
Display				
	1	LCD, 20x4 Zeichen, HD44780-komp.	eBay	291548935509
LEDs				
LED_O	1	LED, 5mm, orange	Reichelt	LED 5MM R OR
LED_R	1	LED, 5mm, rot	Reichelt	LED 5MM RT
LED_Y	1	LED, 5mm, gelb	Reichelt	LED 5MM GE
LED_G	1	LED, 5mm, grün	Reichelt	LED 5MM GN
Quarz				
Q1	1	Standardquarz, Grundton, 9,8304 Mhz	Reichelt	9,8304-HC49U-S
Widerstände				
R13	1	6k8	Reichelt	1/4W 6,8K
R12, R14	2	3k3	Reichelt	METALL 3,30K
R1, R2, R3, R7	4	10k	Reichelt	METALL 10,0K
R4...6, R8...11	7	1k0	Reichelt	METALL 1,00K
Mechanische Bauteile				
X1	1	Sub-D-Buchse, 9-pol	Reichelt	D-SUB BU 09US
SV2	1	Wannenstecker, 10-pol	Reichelt	WSL 10G
	1	Box	Reichelt	GEH KS 50

Bauteil	Anz.	Beschreibung	Händler	Artikelnr.
	1	USB-Kabel als Stromkabel	Reichelt	AK 670/2-1,0
		HF-Komponenten		
	1	SMA-Kabel Funkmodul-Gehäuse	eBay	161134814025
	1	Antenne 868 MHz	eBay	380436601891
	1	SMA-Platinenbuchse, 1,6mm	eBay	220952712009

Tabelle 10.1: Materialliste für den Transmitter

### Materialliste Raspberry-Pi-Transmitter

Bauteil	Anz.	Beschreibung	Händler	Artikelnr.
Kondensatoren				
C1, C3	2	Elektrolytkondensator, 10uF	Reichelt	RAD 10/100
C2, C4, C5, C6	4	Keramikkondensator, 100nF	Reichelt	X7R-2,5 100N
Integrierte Schaltungen				
IC1	1	LM3940-3,3	Reichelt	LM 3940 IT3,3
IC2	1	ULN2003A	Reichelt	ULN 2003A
IC3	1	MCP3202-CI/P	eBay	250878853097
	1	RFM69CW	Pollin	810 303
Einplatinenrechner und Zubehör				
	1	Raspberry Pi 3	Reichelt	RASPBERRY PI 3
	1	Micro-SD-Speicherplatte, 16 GB	Reichelt	INTENSO 3433470
	1	USB-Soundkarte	Amazon	B00C7LXUDY
	1	Powerbank	Amazon	B01KPFC4B2
	1	Waveshare 7 inch 1024*600 Capacitive Touch Screen LCD	Amazon	B015E8EDYQ
LEDs				
LED_O	1	LED, 5mm, orange	Reichelt	LED 5MM R OR
LED_G	1	LED, 5mm, grün	Reichelt	LED 5MM GN
Taster/Schalter				
Schlüsselschalter	1	Miniaturschlüsselschalter	Pollin	420 664
Taster	1	LED, 12mm, 3V, blau	AliExpress	Link
Taster	1	LED, 12mm, 3V, rot	AliExpress	Link
Widerstände				
R_10	1	100R	Reichelt	METALL 100
R_12	1	150R	Reichelt	METALL 150
R1, R8, R9, R10	4	10k	Reichelt	METALL 10,0K
R11	1	2k2	Reichelt	METALL 2,20K
R_7, R_8, R_11	3	1k0	Reichelt	METALL 1,00K
Mechanische Bauteile				
	4	Schraubklemme, 6-pol	Reichelt	AKL 059-06
	1	Buchsenleiste, 2x20-pol	Pollin	451 358
	1	Buchsenleiste mit 6 Plätzen	Reichelt	MPE 094-1-006
	1	Buchsenleiste mit 7 Plätzen	Reichelt	MPE 094-1-007
	1	Micro-USB-Kabel Powerbank-Pi	Reichelt	DELOCK 83897
	1	Micro-USB-Kabel Pi-Touchscreen	AliExpress	Link
	1	HDMI-Kabel Pi-Touchscreen	eBay	361031629625
HF-Komponenten				
	1	SMA-Kabel Funkmodul-Gehäuse	eBay	161134814025
	1	Antenne 868 MHz	eBay	380436601891
	1	SMA-Platinenbuchse, 1,6mm	eBay	220952712009

Tabelle 10.2: Materialliste für den Raspberry-Pi-Transmitter

### Materialiste Zündbox v1

Bauteil	Anz.	Beschreibung	Händler	Artikelnr.
Kondensatoren				
C13, C27, C19	3	Elektrolytkondensator, 100uF	Reichelt	RAD 105 100/35
C14	1	Elektrolytkondensator, 10uF	Reichelt	RAD 10/100
C21, C22	2	Elektrolytkondensator, 4700uF 35V	Reichelt	RAD 4.700/35
C1...7, C15, C16, C20, C23...26	14	Keramikkondensator, 100nF	Reichelt	X7R-2,5 100N
C8, C9	2	Keramikkondensator, 15pF	Reichelt	KERKO 15P
C18	1	Keramikkondensator, 220pF	Reichelt	KERKO 220P
Dioden				
D3	1	1N4002	Reichelt	1N 4002
D2	1	1N5819	Reichelt	1N 5819
Integrierte Schaltungen				
IC2	1	ATMEGA 328P	Reichelt	ATMEGA 328P-PU
IC4	1	MC33063	Reichelt	MC 33063 AP1
IC1	1	MAX202	Reichelt	MAX 202 ECPE
IC6, IC7	2	74HC595	Reichelt	74HC 595
IC8	1	LM1086	Reichelt	LM 1086 IT3,3
U\$4	1	RFM69CW	Pollin	810 303
U\$2	1	DS18B20	Reichelt	DS 18B20
Induktivität				
L1	1	68 uH, stehend	Reichelt	L-07HCP 68μ
LEDs				
LED_O	1	LED, 5mm, orange	Reichelt	LED 5MM R OR
LED_R	1	LED, 5mm, rot	Reichelt	LED 5MM RT
LED_Y	1	LED, 5mm, gelb	Reichelt	LED 5MM GE
LED_G	1	LED, 5mm, grün	Reichelt	LED 5MM GN
LED1...16	16	LED, 3mm, grün	Reichelt	LED 3MM GN
Quarz				
Q1	1	Standardquarz, Grundton, 9,8304 Mhz	Reichelt	9,8304-HC49U-S
MOSFETs				
Q2...Q17	16	IRF3708	AliExpress	Link
Widerstände				
R16	1	180	Reichelt	METALL 180
R17	1	0R22	eBay	221583734560
R1...3, R7, R20, R37	7	10k	Reichelt	METALL 10,0K
R19	1	100k	Reichelt	METALL 100K
R18	1	1M5	Reichelt	METALL 1,50M
R4...6, R8...11	7	1k	Reichelt	METALL 1,00K
R_Z	1	2R2-9W	Reichelt	9W VERT. 2,2
R12, R14, R38	3	3k3	Reichelt	METALL 3,30K
R15	1	56k	Reichelt	METALL 56K
R21...36	16	6k8	Reichelt	1/4W 6,8K
RN1, RN2	2	Network, 9Pin, 10k	Reichelt	SIL 9-8 10K
HF-Komponenten				
	1	SMA-Kabel Funkmodul-Gehäuse	eBay	291548738413
	1	SMA-Kabel Gehäuse-Antenne	eBay	151505370986
	1	Antenne 868 MHz	eBay	380436601891
	1	SMA-Platinenbuchse, 1,6mm	eBay	220952712009
Mechanische Bauteile				
X1	1	Sub-D-Buchse, 9-pol	Reichelt	D-SUB BU 09US
SV2	1	Wannenstecker, 10-pol	Reichelt	WSL 10G
	1	Wippschalter	Pollin	420 697
	1	Miniaturschlüsselschalter	Pollin	420 664

Bauteil	Anz.	Beschreibung	Händler	Artikelnr.
	1	Kunststoffgehäuse 021-002-084	Pollin	460 001
	8	Lautsprecherklemmen	Reichelt	PT 932
	1	Koffer	Amazon	Bilora 545
	1	Akku	Reichelt	WP 1,2-12
	4	Schrauben M3x6	Reichelt	SZK M3X6-200
	16	Schrauben M3x10	Reichelt	SKL M3X10-50
	16	Muttern M3	Reichelt	SK-E M3-100
	2	Akku-Flachstecker	Reichelt	FSH-M1 4,75
	1	Stiftleiste	Reichelt	SL 1X36G 2,54
	1	Buchsenleiste mit 6 Plätzen	Reichelt	MPE 094-1-006
	1	Buchsenleiste mit 7 Plätzen	Reichelt	MPE 094-1-007
	1	Deans-T-Plugs-Paar	Pollin	820 129
Kabel				
	1	Flachbandkabel für LED+Schlüsselsch.	Reichelt	AWG 28-10F 3M
	1	Litzen-Sortiment, 0,5 mm <sup>2</sup> , 5x 5 m	Pollin	800 024
	1	Schrumpfschläuche 1,6mm	Reichelt	SDH 1,6 SW
	1	Schrumpfschläuche 3,2mm	Reichelt	SDH 3,2 SW

Tabelle 10.3: Materialliste für die Zündbox (1. Generation)

### Materialiste Zündbox v2

Bauteil	Anz.	Beschreibung	Händler	Artikelnr.
Kondensatoren				
C13, C27	2	Elektrolytkondensator, 100uF	Reichelt	RAD 105 100/35
C14	1	Elektrolytkondensator, 10uF	Reichelt	RAD 10/100
C21, C22	2	Elektrolytkondensator, 4700uF 35V	Reichelt	RAD 4.700/35
C1...7, C15, C16, C23...26	13	Keramikkondensator, 100nF	Reichelt	X7R-2,5 100N
C8, C9	2	Keramikkondensator, 15pF	Reichelt	KERKO 15P
Dioden				
D3	1	1N4002	Reichelt	1N 4002
Integrierte Schaltungen				
IC2	1	ATMEGA 328P	Reichelt	ATMEGA 328P-PU
IC1	1	MAX202	Reichelt	MAX 202 ECPE
IC6, IC7	2	74HC595	Reichelt	74HC 595
IC8	1	LM1086	Reichelt	LM 1086 IT3,3
U\$4	1	RFM69CW	Pollin	810 303
U\$2	1	DS18B20	Reichelt	DS 18B20
Step-Up-Modul				
—	1	Step-Up M-SU-XL6009	Pollin	351 434
LEDs				
LED_O	1	LED, 5mm, orange	Reichelt	LED 5MM R OR
LED_R	1	LED, 5mm, rot	Reichelt	LED 5MM RT
LED_Y	1	LED, 5mm, gelb	Reichelt	LED 5MM GE
LED_G	1	LED, 5mm, grün	Reichelt	LED 5MM GN
LED1...16	16	LED, 3mm, grün	Reichelt	LED 3MM GN
Quarz				
Q1	1	Standardquarz, Grundton, 9,8304 Mhz	Reichelt	9,8304-HC49U-S
MOSFETs und Transistoren				
Q2...Q17	16	IRF3708	AliExpress	Link
Q19	1	IRF4905	Reichelt	IRF 4905
Q18, T1	2	BC337-40	Reichelt	BC 337-40
T2	1	BC327-40	Reichelt	BC 327-40
Widerstände				
R1...3, R7, R16, R20, R37	7	10k	Reichelt	METALL 10,0K
R19	1	100k	Reichelt	METALL 100K
R18	1	1M5	Reichelt	METALL 1,50M
R4...6, R8...11	7	1k	Reichelt	METALL 1,00K
R_Z	1	2R2-9W	Reichelt	9W VERT. 2,2
R_Z1	1	150R-9W	Reichelt	9W VERT. 150
R12, R38	2	3k3	Reichelt	METALL 3,30K
R13	1	270R	Reichelt	METALL 270
R14	1	1k5	Reichelt	METALL 1,50K
R21...36	16	6k8	Reichelt	1/4W 6,8K
RN1, RN2	2	Network, 9Pin, 10k	Reichelt	SIL 9-8 10K
HF-Komponenten				
	1	SMA-Kabel Funkmodul-Gehäuse	eBay	291548738413
	1	SMA-Kabel Gehäuse-Antenne	eBay	151505370986
	1	Antenne 868 MHz	eBay	380436601891
	1	SMA-Platinenbuchse, 1,6mm	eBay	220952712009
Mechanische Bauteile				
X1	1	Sub-D-Buchse, 9-pol	Reichelt	D-SUB BU 09US
SV2	1	Wannenstecker, 10-pol	Reichelt	WSL 10G
	1	Wippschalter	Pollin	420 697

Bauteil	Anz.	Beschreibung	Händler	Artikelnr.
	1	Miniatur-Schlüsselschalter	Pollin	420 664
	1	Kunststoffgehäuse 021-002-084	Pollin	460 001
	8	Lautsprecherklemmen	Reichelt	PT 932
	1	Koffer	Amazon	Bilora 545
	1	Akku	Reichelt	WP 1,2-12
	4	Schrauben M3x6	Reichelt	SZK M3X6-200
	16	Schrauben M3x10	Reichelt	SKL M3X10-50
	16	Muttern M3	Reichelt	SK-E M3-100
	2	Akku-Flachstecker	Reichelt	FSH-M1 4,75
	1	Stiftleiste	Reichelt	SL 1X36G 2,54
	1	Buchsenleiste mit 6 Plätzen	Reichelt	MPE 094-1-006
	1	Buchsenleiste mit 7 Plätzen	Reichelt	MPE 094-1-007
	1	Deans-T-Plugs-Paar	Pollin	820 129
Kabel				
	1	Flachbandkabel für LED+Schlüsselsch.	Reichelt	AWG 28-10F 3M
	1	Litzen-Sortiment, 0,5 mm <sup>2</sup> , 5x 5 m	Pollin	800 024
	1	Schrumpfschläuche 1,6mm	Reichelt	SDH 1,6 SW
	1	Schrumpfschläuche 3,2mm	Reichelt	SDH 3,2 SW

Tabelle 10.4: Materialliste für die Zündbox (2. Generation)

### Materialiste Zündbox v3

Bauteil	Anz.	Beschreibung	Händler	Artikelnr.
Kondensatoren				
C3, C14, C15	3	Elektrolytkondensator, 10uF	Reichelt	RAD 10/100
C11, C12	2	Elektrolytkondensator, 4700uF 35V	Reichelt	RAD 4.700/35
C1, C2, C4...6, C9, C10, C13, C16...19	12	Keramikkondensator, 100nF	Reichelt	X7R-2,5 100N
C7, C8	2	Keramikkondensator, 15pF	Reichelt	KERKO 15P
Dioden				
D17, D24	2	1N4148	Reichelt	1N 4148
D22, D23	2	1N4001	Reichelt	1N 4001
Integrierte Schaltungen				
U1	1	DM13A	eBay	—
U2	1	MCP6002_IP	Reichelt	MCP 6002-I/P
U3	1	RFM69CW	Pollin	810 303
U4	1	ATMEGA 328P	Reichelt	ATMEGA 328P-PU
U5, U8	2	74HC595	Reichelt	74HC 595
U7	1	LM1086	Reichelt	LM 1086 IT3,3
U9	1	MCP2221_IP	Reichelt	MCP 2221-I/P
U10	1	DS18B20	Reichelt	DS 18B20
Step-Up-Modul				
U6	1	Step-Up M-SU-XL6009	Pollin	351 434
LEDs				
D19	1	LED, 5mm, orange	Reichelt	LED 5MM R OR
D20	1	LED, 5mm, rot	Reichelt	LED 5MM RT
D18	1	LED, 5mm, gelb	Reichelt	LED 5MM GE
D21	1	LED, 5mm, grün	Reichelt	LED 5MM GN
D1...16	16	LED, 3mm, grün	Reichelt	LED 3MM GN
Quarz				
Y1	1	Standardquarz, Grundton, 9,8304 MHz	Reichelt	9,8304-HC49U-S
MOSFETs und Transistoren				
Q1...16	16	IRF3708	AliExpress	Link
Q19	1	IRF4905	Reichelt	IRF 4905
Q18, Q21	2	BC337-40	Reichelt	BC 337-40
Q17, Q20	2	BC327-40	Reichelt	BC 327-40
Widerstände				
R1	1	47k	Reichelt	METALL 47,0K
R2	1	150	Reichelt	METALL 150
R3	1	150k	Reichelt	METALL 150K
R4	1	180k	Reichelt	METALL 180K
R5, R7, R16	3	100k	Reichelt	METALL 100K
R6	1	120k	Reichelt	METALL 120K
R8, R9, R15, R17,	6	10k	Reichelt	METALL 10,0K
R27, R29				
R10...13, R18...20	7	1k	Reichelt	METALL 1,00K
R14	1	1M5	Reichelt	METALL 1,50M
R21, R23...25, R30	5	3k3	Reichelt	METALL 3,30K
R22	1	1k5	Reichelt	METALL 1,50K
R26	1	2R2-9W	Reichelt	9W VERT. 2,2
R28	1	270R	Reichelt	METALL 270
RN1, RN2	2	Network, 9Pin, 10k	Reichelt	SIL 9-8 10K
HF-Komponenten				
	1	SMA-Kabel Funkmodul-Gehäuse	eBay	291548738413
	1	SMA-Kabel Gehäuse-Antenne	eBay	151505370986
	1	Antenne 868 MHz	eBay	380436601891

Bauteil	Anz.	Beschreibung	Händler	Artikelnr.
	1	SMA-Platinenbuchse, 1,6mm	eBay	220952712009
Mechanische Bauteile				
J1	1	USB-Buchse Typ B	Reichelt	USB BW
J2	1	Wannenstecker, 10-pol	Reichelt	WSL 10G
	1	Wippschalter	Pollin	420 697
	1	Miniatur-Schlüsselschalter	Pollin	420 664
	1	Kunststoffgehäuse 021-002-084	Pollin	460 001
	8	Lautsprecherklemmen	Reichelt	PT 932
	1	Koffer	Amazon	Bilora 545
	1	Akku	Reichelt	WP 1,2-12
	4	Schrauben M3x6	Reichelt	SZK M3X6-200
	16	Schrauben M3x10	Reichelt	SKL M3X10-50
	16	Muttern M3	Reichelt	SK-E M3-100
	2	Akku-Flachstecker	Reichelt	FSH-M1 4,75
	1	Stiftleiste	Reichelt	SL 1X36G 2,54
	1	Buchsenleiste mit 6 Plätzen	Reichelt	MPE 094-1-006
	1	Buchsenleiste mit 7 Plätzen	Reichelt	MPE 094-1-007
	1	Deans-T-Plugs-Paar	Pollin	820 129
Kabel				
	1	Flachbandkabel für LED+Schlüsselsch.	Reichelt	AWG 28-10F 3M
	1	Litzen-Sortiment, 0,5 mm <sup>2</sup> , 5x 5 m	Pollin	800 024
	1	Schrumpfschläuche 1,6mm	Reichelt	SDH 1,6 SW
	1	Schrumpfschläuche 3,2mm	Reichelt	SDH 3,2 SW

Tabelle 10.5: Materialliste für die Zündbox (3. Generation)

### Materialiste Zündbox v3 mit 32 Kanälen

Bauteil	Anz.	Beschreibung	Händler	Artikelnr.
<b>Kondensatoren</b>				
C3, C14, C15	3	Elektrolytkondensator, 10uF	Reichelt	RAD 10/100
C11, C12	2	Elektrolytkondensator, 4700uF 35V	Reichelt	RAD 4.700/35
C1, C2, C4...6, C9, C10, C13, C16...22	15	Keramikkondensator, 100nF	Reichelt	X7R-2,5 100N
C7, C8	2	Keramikkondensator, 15pF	Reichelt	KERKO 15P
<b>Dioden</b>				
D33, D40	2	1N4148	Reichelt	1N 4148
D38, D39	2	1N4001	Reichelt	1N 4001
<b>Integrierte Schaltungen</b>				
U1, U11	2	DM13A	eBay	—
U2	1	MCP6002_IP	Reichelt	MCP 6002-I/P
U3	1	RFM69CW	Pollin	810 303
U4	1	ATMEGA 328P	Reichelt	ATMEGA 328P-PU
U5, U8, U12, U13	4	74HC595	Reichelt	74HC 595
U7	1	LM1086	Reichelt	LM 1086 IT3,3
U9	1	MCP2221_IP	Reichelt	MCP 2221-I/P
U10	1	DS18B20	Reichelt	DS 18B20
<b>Step-Up-Modul</b>				
U6	1	Step-Up M-SU-XL6009	Pollin	351 434
<b>LEDs</b>				
D35	1	LED, 5mm, orange	Reichelt	LED 5MM R OR
D36	1	LED, 5mm, rot	Reichelt	LED 5MM RT
D34	1	LED, 5mm, gelb	Reichelt	LED 5MM GE
D37	1	LED, 5mm, grün	Reichelt	LED 5MM GN
D1...32	32	LED, 3mm, grün	Reichelt	LED 3MM GN
<b>Quarz</b>				
Y1	1	Standardquarz, Grundton, 9,8304 Mhz	Reichelt	9,8304-HC49U-S
<b>MOSFETs und Transistoren</b>				
Q1...32	32	IRF3708	AliExpress	Link
Q35	1	IRF4905	Reichelt	IRF 4905
Q34, Q37	2	BC337-40	Reichelt	BC 337-40
Q33, Q36	2	BC327-40	Reichelt	BC 327-40
<b>Widerstände</b>				
R1, R31	2	47k	Reichelt	METALL 47,0K
R2	1	150	Reichelt	METALL 150
R3	1	150k	Reichelt	METALL 150K
R4	1	180k	Reichelt	METALL 180K
R5, R7, R16	3	100k	Reichelt	METALL 100K
R6	1	120k	Reichelt	METALL 120K
R8, R9, R15, R17,	8	10k	Reichelt	METALL 10,0K
R27, R29, R32, R33				
R10...13, R18...20	7	1k	Reichelt	METALL 1,00K
R14	1	1M5	Reichelt	METALL 1,50M
R21, R23...25, R30,	6	3k3	Reichelt	METALL 3,30K
R34				
R22	1	1k5	Reichelt	METALL 1,50K
R26	1	2R2-9W	Reichelt	9W VERT. 2,2
R28	1	270R	Reichelt	METALL 270
RN1, RN2, RN3,	4	Network, 9Pin, 10k	Reichelt	SIL 9-8 10K
RN4				
<b>HF-Komponenten</b>				
1		SMA-Kabel Funkmodul-Gehäuse	eBay	291548738413

Bauteil	Anz.	Beschreibung	Händler	Artikelnr.
	1	SMA-Kabel Gehäuse-Antenne	eBay	151505370986
	1	Antenne 868 MHz	eBay	380436601891
	1	SMA-Platinenbuchse, 1,6mm	eBay	220952712009
Mechanische Bauteile				
J1	1	USB-Buchse Typ B	Reichelt	USB BW
J2	1	Wannenstecker, 10-pol	Reichelt	WSL 10G
	1	Wippschalter	Pollin	420 697
	1	Miniatur-Schlüsselschalter	Pollin	420 664
	16	Lautsprecherklemmen	Reichelt	PT 932
	1	Koffer	Amazon	Bilora 545
	1	Akku	Reichelt	WP 1,2-12
	4	Schrauben M3x6	Reichelt	SZK M3X6-200
	32	Schrauben M3x10	Reichelt	SKL M3X10-50
	32	Muttern M3	Reichelt	SK-E M3-100
	2	Akku-Flachstecker	Reichelt	FSH-M1 4,75
	1	Stiftleiste	Reichelt	SL 1X36G 2,54
	1	Buchsenleiste mit 6 Plätzen	Reichelt	MPE 094-1-006
	1	Buchsenleiste mit 7 Plätzen	Reichelt	MPE 094-1-007
	1	Deans-T-Plugs-Paar	Pollin	820 129
Kabel				
	1	Flachbandkabel für LED+Schlüsselsch.	Reichelt	AWG 28-10F 3M
	1	Litzen-Sortiment, 0,5 mm <sup>2</sup> , 5x 5 m	Pollin	800 024
	1	Schrumpfschlüsse 1,6mm	Reichelt	SDH 1,6 SW
	1	Schrumpfschlüsse 3,2mm	Reichelt	SDH 3,2 SW

Tabelle 10.6: Materialliste für die Zündbox (3. Generation) mit 32 Kanälen



## 11 Platinenherstellung

Wer zur Platinenherstellung nicht auf die Dienste eines PCB-Herstellers zurückgreifen will, findet in Abbildung 11.1 die Platine des Transmitters (Platinenabmessungen: 91,44 mm × 66,04 mm) als Druckvorlage für den Tonertransfer<sup>8</sup> bzw. als Belichtungsvorlage. In Abbildung 11.2 findet sich das Layout der Zündbox (Platinenabmessungen: 157,48 mm × 97,79 mm bei den 16-Kanal-Versionen und 214 mm × 128 mm bei der 32-Kanal-Version) und in Abbildung 11.3 das der Adapterplatine zum Auflöten des Funkmoduls (Abmessungen einer einzelnen Adapterplatine: 31,75 mm × 22,86 mm).

Die Oberseite ist hierbei jeweils schon gespiegelt, die Ausdrücke können für einen Tonertransfer also einfach ausgedruckt und im Zwischenraum gefaltet werden, wobei auf möglichst exakte Deckung zu achten ist.

Die Layouts der späteren Zündboxgenerationen und der Raspberry-Pi-Aufsteckplatine sind aufgrund ihrer Komplexität nicht als Tonertransfer-Vorlagen verfügbar, hier kann auf einen der mittlerweile relativ günstigen PCB-Fertiger zurückgegriffen werden. Für Selbstfertigung kann man die Fertigungsdaten den jeweiligen Gerberdateien entnehmen.

---

<sup>8</sup>Eine Einführung zum Ätzen mit dieser Methode gibt es [HIER](#)

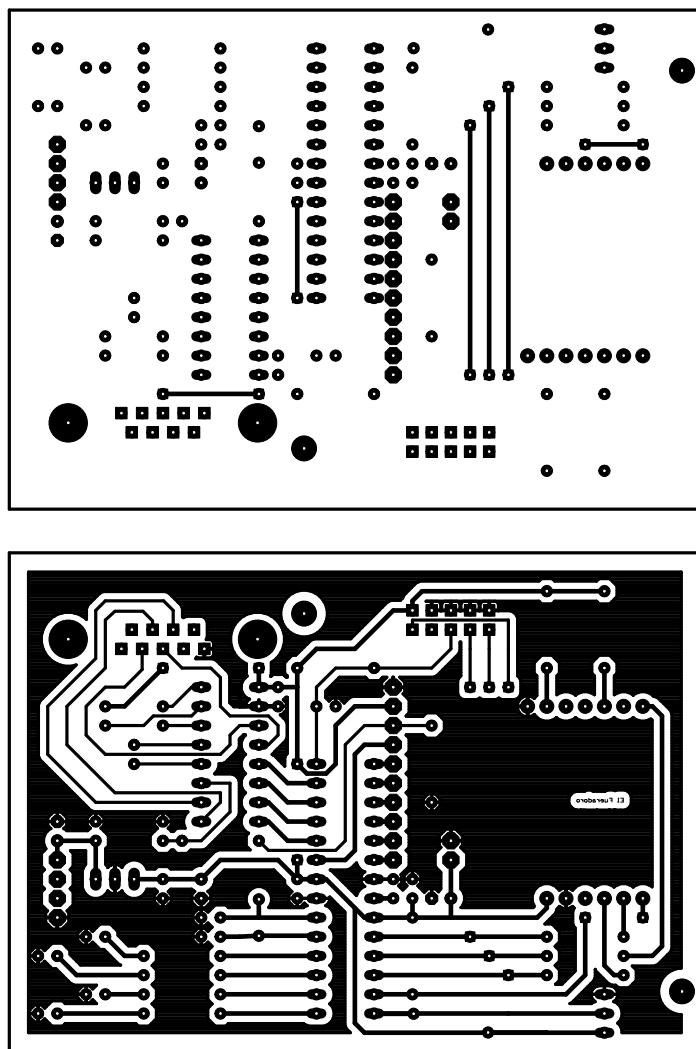


Abbildung 11.1: Ober- und Unterseite des Transmitters für Toner-Transfer-Verfahren/Belichtung

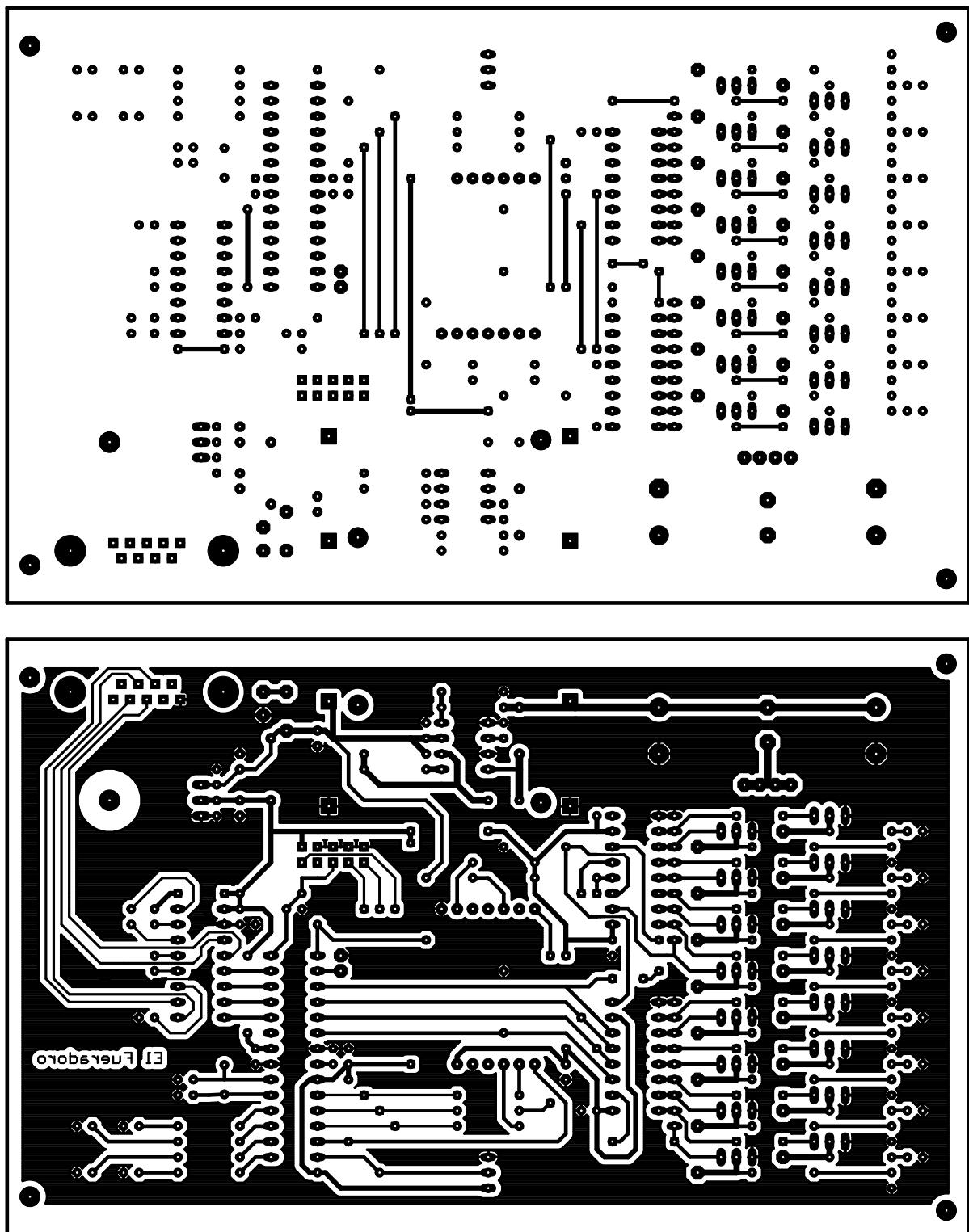


Abbildung 11.2: Ober- und Unterseite der Zündbox für Toner-Transfer-Verfahren/Belichtung

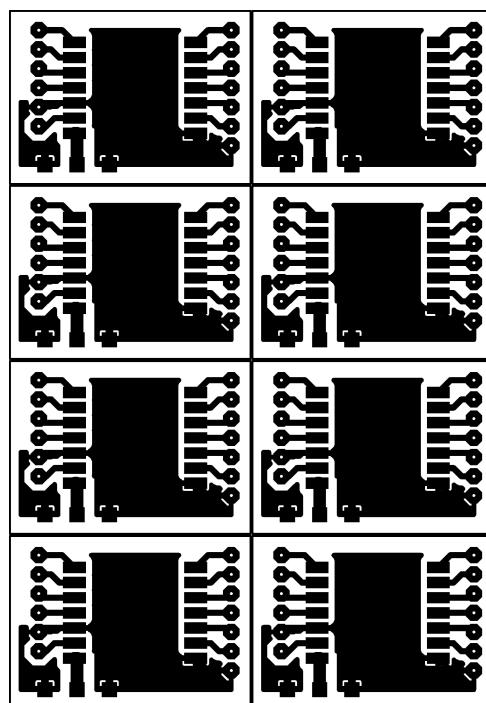


Abbildung 11.3: Adapterplatine für Funkmodule (8 Stück) für Toner-Transfer-Verfahren/Belichtung

# 12 Aufbau

## 12.1 PC-Transmitter und Zündbox

Der Aufbau umfasst alle Schritte von der blanken Platine hin zum fertigen Gehäuse/Koffer. Hierfür sind verschiedene handwerkliche Tätigkeiten, vor allem das Elektroniklöten<sup>9</sup>, aber auch das Bohren, Schneiden, Kleben und evtl. Trennen mittels Trennscheibe und Feilen nötig.

Im eigenen Interesse ist darauf zu achten, diese Arbeiten sorgfältig und unter Einhaltung der gängigen Sicherheitsregeln durchzuführen. **Beim Bohren und Trennen Schutzbrille tragen!** Beim Löten ist auf richtige Orientierung aktiver Bauteile (Dioden, Elektrolytkondensatoren, Temperatursensor, Integrierte Schaltungen) sowie Anschlusskabel zu achten, vor dem Einschalten soll die eigene Arbeit auch kritisch auf beim Löten entstandene Kurzschlüsse getestet werden<sup>10</sup>. Eine Laborspannungsquelle mit einstellbarer Strombegrenzung – oder ein Steckernetzteil mit maximal 2 A Ausgangsstrom – zu Testzwecken tun hier gute Dienste, wobei auf die Einhaltung der zulässigen Betriebsspannungen (Transmitter 5-7 V, Zündbox 8-15 V) zu achten ist.

### 12.1.1 Kabel

Zur Verbindung der Platinen mit der Peripherie wird bei *El Fueradoro* eine Vielzahl von Kabeln benötigt, die grob in drei Kategorien unterteilt werden können:

1. Flachbandkabel bzw. Flachbandkabel-Adern zum Anschluss von LCD (außer Hintergrundbeleuchtung), LEDs und Schlüsselschalter
2. Litze mit einer Querschnittsfläche von mindestens  $0,5 \text{ mm}^2$ , d. h. einem Mindestdurchmesser von 0,8 mm, zum Anschluss von LCD-Hintergrundbeleuchtung, Netzschatzer und Zündklemmen
3.  $50 \Omega$ -Koaxialkabel als Antennenkabel

Die Koaxialkabel werden in diesem Abschnitt nicht behandelt, da dafür oft Spezialwerkzeug notwendig ist und davon ausgegangen wird, dass diese Kabel bereits fertig konfektioniert erworben werden. Für

---

<sup>9</sup>Ein gutes – wenn auch englischsprachiges – Löt-Tutorial mit wichtigen Grundlagen gibt es [HIER](#)

<sup>10</sup>Hier klicken, um zu sehen, was ein kurzgeschlossener Blei-Gel-Akku mit Drähten/Leiterbahnen anstellt, sofern er nicht direkt explodiert!

Device (T/Z)	Art	Adern	Länge	Anzahl	Verwendungszweck
Z	Litze	–	210 mm	4	Rote Klemmen (spaltenweise)
Z	Litze	–	210 mm	16	Schwarze Klemmen
Z	Litze	–	200 mm	2	Netzschalter
Z	Litze	–	250 mm	4	Batterie (2x rot, 2x schwarz)
Z	Flachband	2	200 mm	16	Kanal-LEDs
Z	Flachband	2	200 mm	4	Status-LEDs
Z	Flachband	2	200 mm	1	Schlüsselschalter
T	Litze	–	100 mm	2	LCD-Hintergrundbeleuchtung
T	Flachband	2	100 mm	4	Status-LEDs
T	Flachband	10	100 mm	1	LCD

Tabelle 12.1: Übersicht über benötigte Kabelverbindungen

die Konfektionierung der anderen Kabel gilt, dass diese so kurz wie möglich aber gleichzeitig auch so lang wie nötig sein sollten, um das Verlöten/Verkleben annehmbar zu gestalten und das Gehäuse später noch einmal öffnen zu können, ohne gleich alles abzureißen.

In Tabelle 12.1 sind die benötigten Abschnitte aufgelistet. Alle Kabel sollten am einen Ende jeweils auf einer Länge von 4 mm zum Festlöten an der Platine, am anderen auf einer Länge von 6 mm abisoliert und verzinnt werden. Zur Vereinfachung der Arbeit ist es ratsam, die Adern des Flachbandkabels erst danach zu trennen, so dass nicht jeder „Zweierverbund“ einzeln abisoliert und verzinnt werden muss.

Es dient der Übersichtlichkeit und dem späteren Verständnis ungemein, wenn man verschiedene Kabelfarben verwendet und sich dabei an gängige Konventionen hält (Akkuspannung rot, Masse schwarz).

### 12.1.2 Platinen

Ausgangspunkt der Bestückung ist die geätzte und gebohrte Platine mit allen Leiterbahnen auf der Ober- und Unterseite. Für selbst gefertigte, einseitig geätzte Platinen, bei denen die Leitungen auf der Oberseite als Drahtbrücken ausgeführt sind, sind diese in Abbildung 12.1 dargestellt.

Anschließend sollten die Bauteile in folgender Reihenfolge eingelötet werden:

**Widerstände** mit Ausnahme von R\_Z, bei den Widerstandsnetzwerken auf korrekte Orientierung achten

**Dioden** mit korrekter Polung

**Keramikkondensatoren**

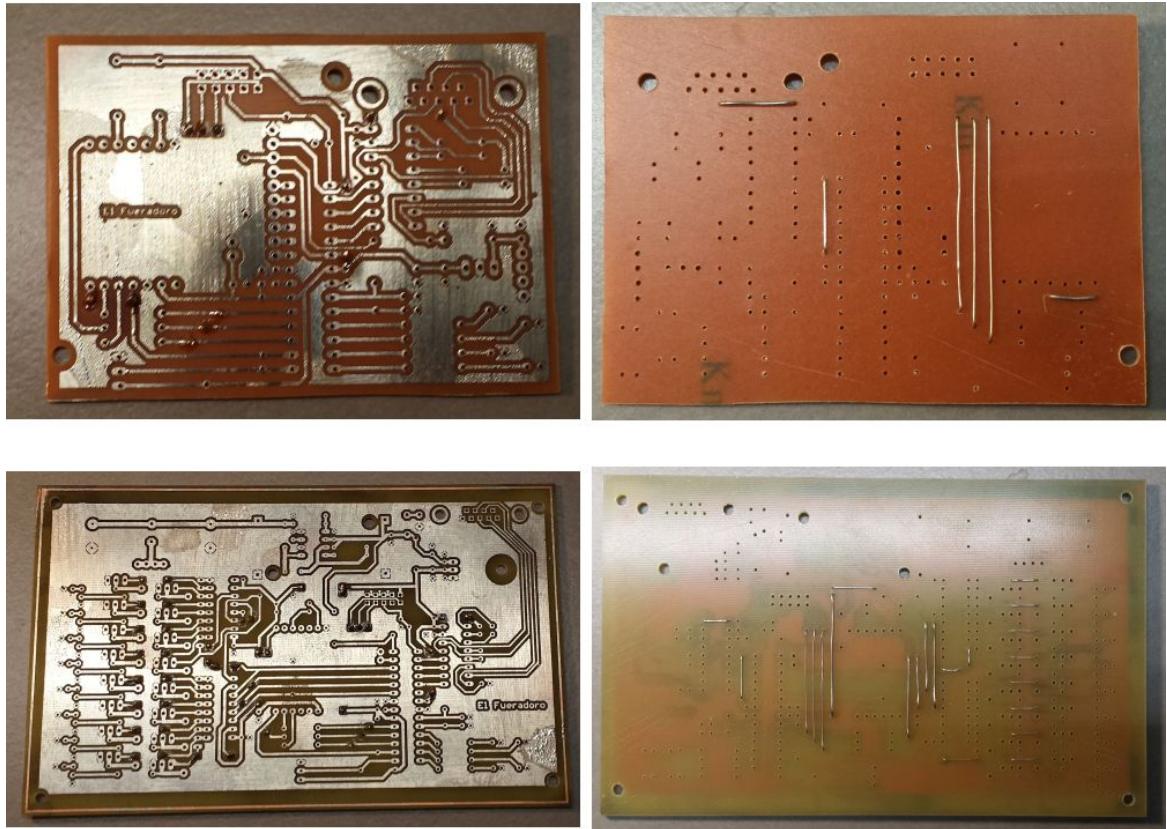


Abbildung 12.1: Platinen für Transmitter (oben) und Zündbox (unten)

### Quarz

**Buchsenleisten** für Funkmoduladapter

**Elektrolytkondensatoren** Polung beachten und C14 beim Transmitter waagrecht legen!

**Leistungswiderstand R\_Z**

**Temperatursensor** mit korrekter Orientierung

**Integrierte Schaltungen** mit korrekter Orientierung

Der Stand bis zu diesem Punkt ist in Abbildung 12.2 dargestellt.

**Anschluss-Kabel** für Klemmen (Zündbox) bzw. LCD (Transmitter)

**Schalterkabel** für Netz- und Schlüsselschalter

**LED-Kabel** für Status-LEDs und Kanal-LEDs. Um den Überblick zu behalten sollte dabei jeweils

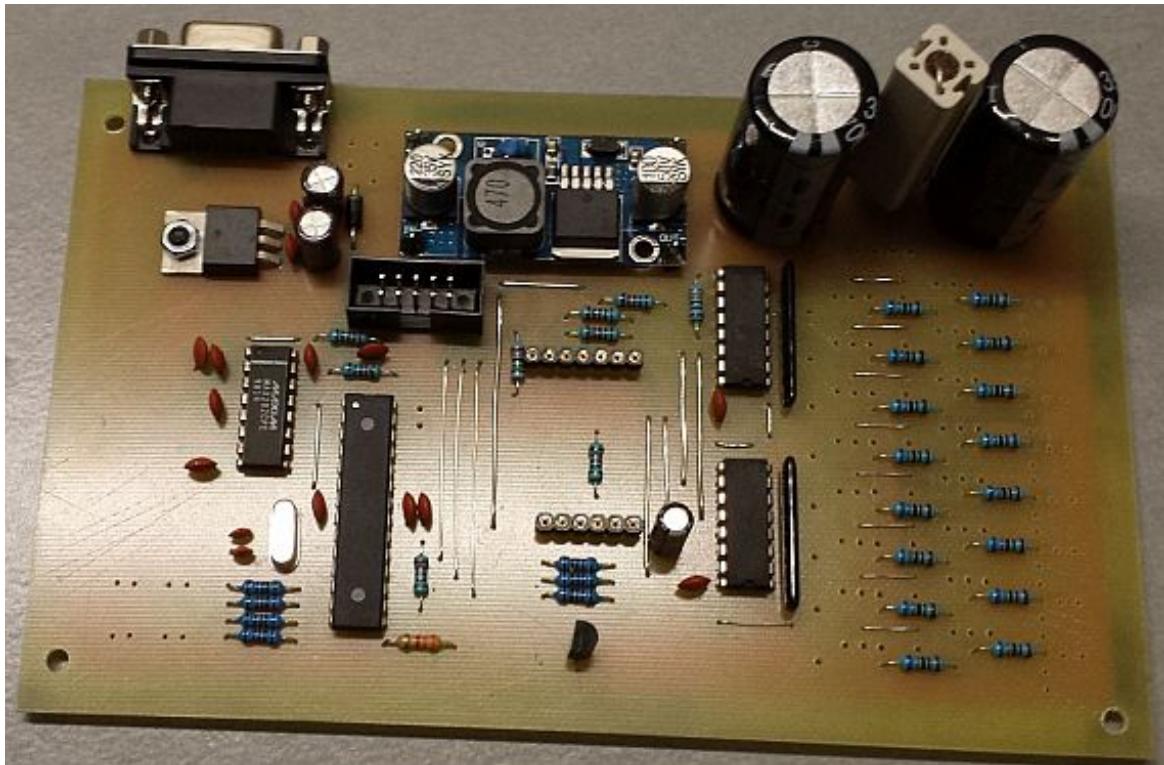


Abbildung 12.2: Zündboxplatine (1. Generation) vor dem Einlöten der Kabel und MOSFETs

das Kabel für den GND-Anschluss vor dem Einlöten markiert werden

**MOSFETs** mit korrekter Orientierung (Metallplatte sitzt auf der Seite der großen Elkos)

**Batterie- bzw. USB-Versorgungs-Kabel** Falls bei der Zündbox fertig konfektionierte Kabel mit Anschluss verwendet werden, vor dem Löten die Seitenwand auf der Schalterseite durchbohren und beide Kabel durchfädeln. Beim Trasmitter wird das Kabel zwischen zwei miteinander verschraubten Teilen verlegt.

## 12.2 Raspberry-Pi-Aufsteckplatine

Das Einlöten der Bauteile erfolgt in bewährter Reihenfolge:

**Widerstände**

**Keramikkondensatoren**

**Einreihige Buchsenleisten** für Funkmoduladapter

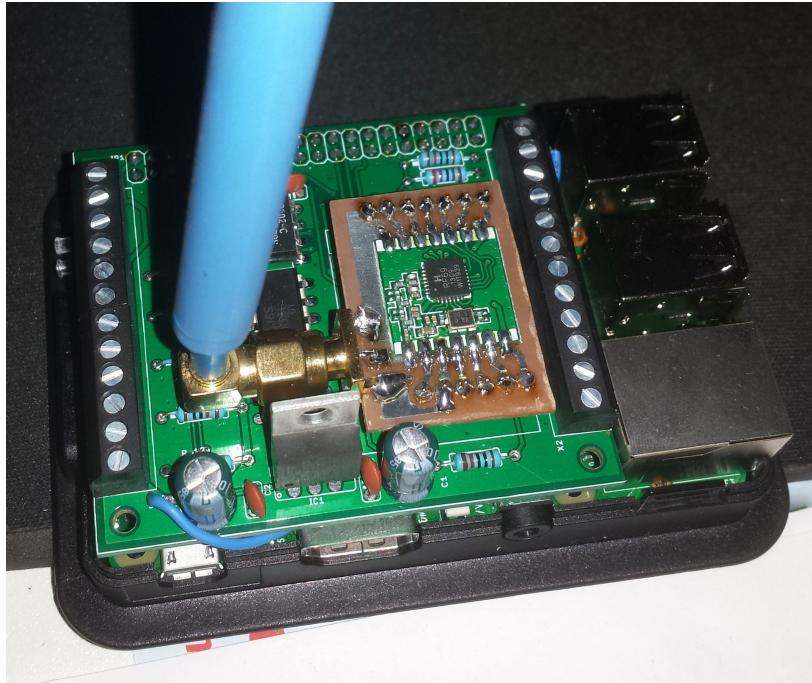


Abbildung 12.3: Raspberry-Pi-Erweiterungsplatine mit aufgestecktem Zündmodul und Antenne

### Schraubklemmen

**Elektrolytkondensatoren** Polung beachten!

**Integrierte Schaltungen** mit korrekter Orientierung

**Zweireihige Buchsenleiste** ACHTUNG: Buchsenleiste befindet sich auf der Unterseite der Platine, Lötseite ist also hier ausnahmsweise die Oberseite

**Verbindungskabel** zwischen Spannungsmesspunkt und Klemmeneingang

Die fertig bestückte Platine ist in Abbildung 12.3 gezeigt.

## 12.3 Funkmodul-Adapter

Nachdem die Platinen nun fertig bestückt sind, folgen die Schritte für die Fertigstellung des Funkmodul-Adapters:

**Funkmodul** mit allen 14 Anschlüssen (Orientierung beachten) auf der Adapterplatine festlöten. Dabei beachten, dass auf die SMD-Pads auf beiden Seiten etwa gleich weit unter den Anschlüssen hervorstehen und das Modul nicht nach oben oder unten verschoben ist. Am besten mit dem

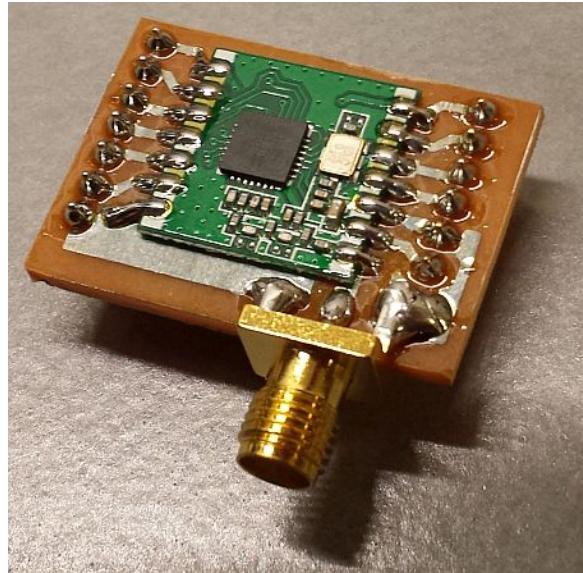


Abbildung 12.4: Fertig aufgebauter Funkmoduladapter

mittleren Pin einer Seite beginnen und das Modul beim Löten korrekt positionieren, dann alle anderen 13 Anschlüsse löten.

**SMA-Buchse** anlöten, so dass sie direkt an der Platine anliegt. Falls Anschlüsse zu lang sind und am Modul anstoßen, vorsichtig mit Trennscheibe (Außenleiter) oder Seitenschneider (Innenleiter) kürzen.

**Stiftleiste** in einen 6-poligen und einen 7-poligen Abschnitt teilen und anlöten.

Der fertige Adapter ist in Abbildung 12.4 gezeigt.

## 12.4 Peripherie

### 12.4.1 Transmitter

Als Gehäuse für den Transmitter wird das schwarze Kunststoffgehäuse GEH KS 50 aus dem Sortiment von Reichelt verwendet. Es besteht aus zwei miteinander zu verschraubenden Teilen, wobei der dünne Teil, auf dem später die Platine befestigt wird, als Rückwand dient. Dementsprechend sind die Bezeichnungen in der folgenden Beschreibung zu verstehen.

Die Vorderseite des Gehäuses muss zunächst mit einem Ausschnitt von 97 mm × 39,5 mm und vier 3 mm-Bohrungen für die Anbringung des LCD versehen werden. Die Abstände zueinander sind in Abbildung 12.5 verdeutlicht. Wer nicht auf seine Messkünste vertrauen möchte oder ein LCD mit

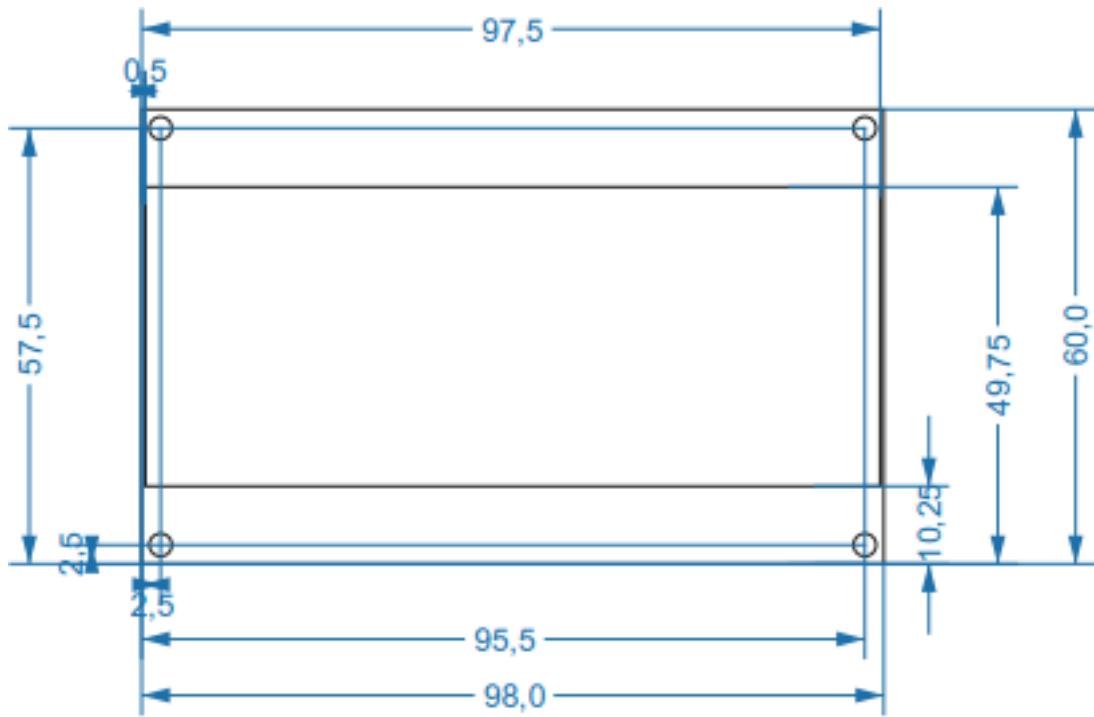


Abbildung 12.5: Abmessungen (in mm) des LCD

abweichenden Abmessungen besitzt, sollte zunächst einen passenden Ausschnitt für den Bildschirm mittig in der Fläche anbringen, anschließend können das LCD aufgelegt und die vier Bohrlöcher markiert und gebohrt werden.

Bevor man das LCD einschraubt, muss noch eine Aussparung für den Sub-D-Anschluss, ein Loch für den Antennenanschluss (6,5 mm), vier Löcher für die Status-LEDs (5 mm) und eine Kerbe für die Versorgungskabeldurchführung in die Wand, welche später als Oberseite des Gehäuses dient, eingebracht werden.

Hierfür sollte man zunächst die fertig bestückte Platine mittels zweier Schrauben in ihrer finalen Position festschrauben und das Oberteil so anlegen, dass die Position der Aussparung für die Sub-D-Buchse angezeichnet werden kann. Die Abmessung der Aussparung sollte  $31 \text{ mm} \times 12,5 \text{ mm}$  betragen. Die Positionen der restlichen Löcher sind aufgrund der Kabelverbindung mit der Platine unkritisch, die ungefähre Lage kann Abbildung 1.2a auf Seite 13 entnommen werden. Das Versorgungskabel mit USB-Stecker wird zwecks Zugentlastung zwischen den beiden zu verschraubenden Teilen eingeklemmt, die Kerbe sollte daher nicht allzu groß ausgeführt werden. Die Bohrarbeiten am Transmitter sind damit erledigt!

Nun müssen noch die elektrischen Verbindungen zwischen Platine und Peripherie hergestellt und die Peripherieteile anschließend befestigt werden – zuerst die vier Status-LEDs:

1. Die Verbindung der Adern des Flachbandkabels auf einer Länge von etwa 35 mm auftrennen

2. Auf jede Ader einen dünnen Schrumpfschlauch der Länge 15 mm stecken (noch nicht erhitzen!)
3. Das kürzere Anschlussbein der LED (Kathode) mit dem Seitenschneider auf eine Länge von 8 mm trimmen und die mit GND verbundene Ader anlöten
4. Das längere Anschlussbein der LED (Anode) mit dem Seitenschneider auf eine Länge von 8 mm trimmen und die andere Ader anlöten
5. Schrumpfschläuche bis ans LED-Gehäuse vorschieben und per Heißluft schrumpfen

Anschließend das Flachbandkabel für das LCD vorbereiten, d. h. abisolieren, verzinnen und Verbindungen soweit lösen, dass alle Anschlüsse bequem erreicht werden können. Da das LCD im 4-Bit-Modus betrieben wird, werden nur die Pins 1-6 sowie 11-16 angeschlossen, 7-10 bleiben offen. Bei Anschluss der Pins 15 und 16 darauf achten, Anode und Kathode nicht zu vertauschen; die Belegung ist in der Regel so, dass die Anode an Pin 15 herausgeführt ist, kann aber von LCD zu LCD variieren. Der Lötkolben kann danach ausgeschaltet werden, jetzt geht es an die Befestigung.

Zunächst wird das LCD am Gehäuse festgeschraubt, wobei darauf zu achten ist, dass die Oberseite auch in die Richtung von Antennen- und Sub-D-Anschluss zeigt. Anschließend die LEDs um den Gehäusering mit Sekunden- oder Heißkleber bestreichen und danach für einige Sekunden fest ins dafür vorgesehene Loch pressen. Nun die SMA-Buchse fest am Gehäuse anschrauben und das andere Kabelende mit der Buchse am Funkmodul-Adapter verbinden.

Unter möglichst geringer Torsion sollte dann der Funkmoduladapter in die Buchsenleisten auf der Platine gesteckt werden.

Nun muss man noch die Antenne anschrauben. Wenn der Bootloader sich bereits auf dem Controller befindet, kann man das Gehäuse zuschrauben. Ansonsten den Transmitter mit Energie versorgen und den Bootloader wie in Abschnitt 13 beschrieben flashen. Jetzt ist der Transmitter fertig aufgebaut und kann zugeschraubt werden! Die Firmware kann wie in Abschnitt 3.3 beschrieben über die serielle Schnittstelle aufgespielt werden, wobei beim ersten Mal noch die Angabe des Dateinamens nötig ist.

### 12.4.2 Zündbox

Abbildung 12.6 zeigt die Bohrschablone für die Oberseite des Kunststoffgehäuses der Zündbox (Kunststoffgehäuse 021-002-084 von Pollin). Diese kann dazu verwendet werden, eine Schablone aus Sperrholz oder Metall anzufertigen, welche später auf die Boxenoberseite gelegt wird, um die nötigen Bohrungen vorzunehmen. Die Oberseite des Gehäuses ist dabei der Teil ohne sichtbare Schraublöcher.

Es ist beim Bohren der Box auf die richtige Orientierung der Schablone zu achten, da die Befestigungsbohrungen für die Platine nicht symmetrisch sind (Schrauben auf der linken Seite, wo sich die serielle

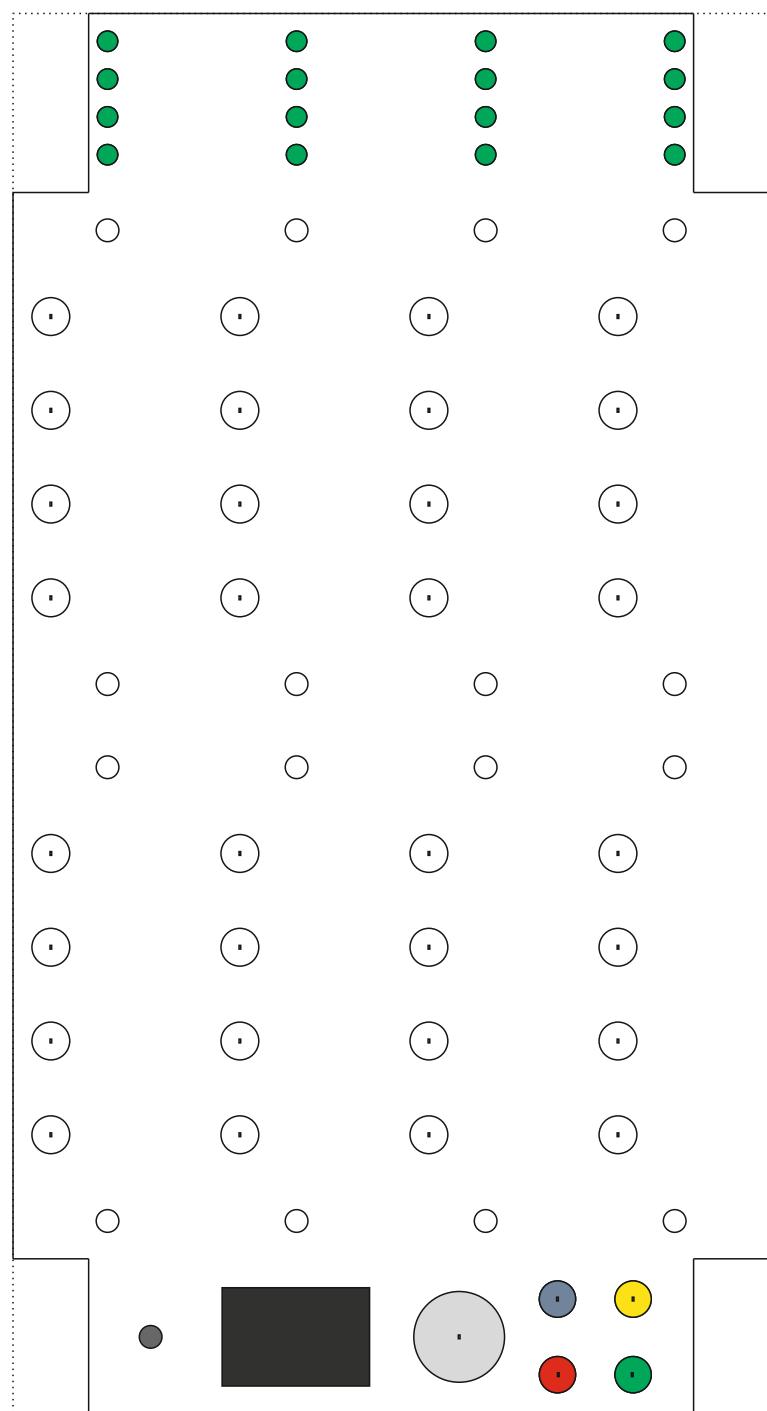


Abbildung 12.6: Bohrschablone für Zündboxoberseite

Schnittstelle befindet, sind enger zusammen als die auf der rechten Seite mit den MOSFETs und Kanal-LEDs) und die Gehäuseenteile durch ein Nut-Feder-System nur in einer Kombination aufeinander gesteckt werden können. Man beachte hierzu Abbildung 12.7.

Damit die Löcher nicht mit Schraubenhalterungen in den Ecken interferieren, sollte zunächst im Gehäuseinneren der passende Ort für die Kanal-LEDs von Kanal 4 und 16 (äußerste grüne LEDs in der obersten und untersten Reihe, Abstand 75 mm) gesucht, die beiden Löcher mit einem 3 mm-Bohrer gebohrt und die Schablone auf der Oberseite in diesen beiden Bohrungen befestigt werden.

Die Bohrlöcher für die 16 Kanal-LEDs sollten mit einem 3-mm-Bohrer, die der vier Status-LEDs mit einem 5 mm-Bohrer ausgeführt werden. Die LEDs werden später (nach dem Verkabeln und Festlöten des Kabels auf der Platine) seitlich mit Heißkleber bestrichen von unten bis zum Anschlag in diese Löcher eingeschoben.

Da die Lautsprecherklemmen mit M3-Gewindeschrauben befestigt werden, wäre der ideale Bohrdurchmesser für die 16 Schraubenlöcher 3,2 mm. Sollte dieser Durchmesser nicht vorhanden sein, kann aber auch mit 3,5 mm oder 3 mm gearbeitet werden. Für die Lötfahnen der Klemmen ist eine rechteckige Aussparung von 4,5 mm × 2 mm nötig, als schnelle Lösung kann auch jeweils ein 5 mm-Loch durch den Mittelpunkt (Diagonalenschnittpunkt) dieser Flächen gebohrt werden.

Die rechteckige Aussparung für den Netzschatz sollte die Größe 19 mm × 13 mm besitzen, der Schlüsselschalter hat einen Einbaudurchmesser von 12 mm und der Durchsteckplatz für die SMA-Buchse sollte mit 6,5 mm vorgebohrt werden. Im nächsten Schritt werden Schlüsselschalter, Klemmen und Netzschatz am Gehäuse verschraubt bzw. eingeklickt. Die Innenansicht des Zündboxdeckels (noch ohne SMA-Anschluss) ist in Abbildung 12.7 dargestellt. Hierbei ist auch zu erkennen, wo sich die Nut (unten, bei den Status-LEDs) und wo die Feder (oben) beim Deckel befinden muss. Dementsprechend muss es bei der Unterseite andersherum sein.

In die linke Einschub-Seitenwand muss, sofern noch nicht im Rahmen des Lötens geschehen, ein Loch für die Durchführung der Batteriekabel mit einer Größe je nach Kabeldurchmesser.

Um die Position der D-Sub-Buchse in der Wand des Unterteils, welche in der Nut endet, zu finnden, muss die Platine möglichst identisch zu ihrer späteren Position ins Gehäuse eingelegt werden, die Aussparung beträgt wie beim Transmitter 31 mm × 12,5 mm. In der dritten Zündboxgeneration kommt statt der D-Sub-Buchse ein USB-Anschluss Typ B zum Einsatz, hier fällt der Ausschnitt entsprechend anders aus. Wenn die Aussparung fertig ist, kann die Platine in ihre endgültige Position gebracht und mit vier Schrauben befestigt werden. Die linke Seitenwand bleibt noch ausgesteckt. Jetzt ist wieder der Lötkolben dran!

Zunächst vier Drahtstücke á 80 mm abschneiden und durch die übereinander liegenden Lötfahnen der roten Klemmen ziehen und mit den Lötfahnen verlöten.

Die Kanal- und Status-LEDs wie auf Seite 109 beschrieben mit den zugehörigen Flachbandkabeln verbinden, anschließend den Schlüssel- und den Netzschatz mit den zugehörigen Anschlusskabeln.

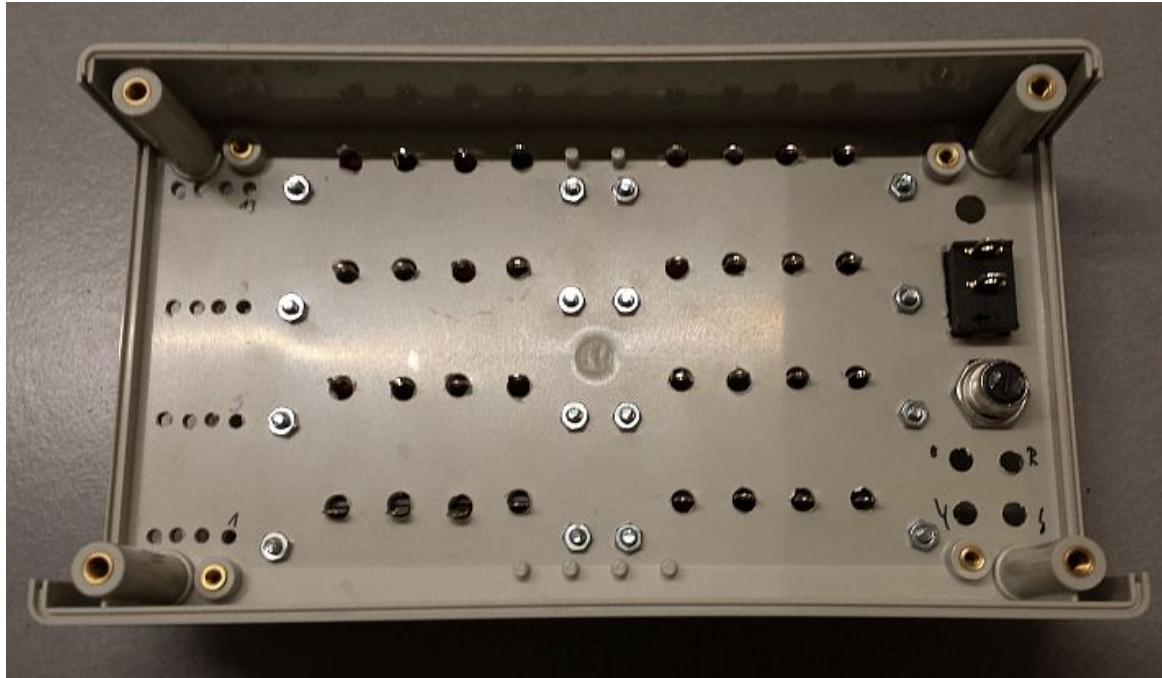


Abbildung 12.7: Innenansicht des Zündboxdeckels

Bei den Schritten in den folgenden Absätzen ist Sorgfalt geboten, da die Kanäle und LEDs richtig zugeordnet werden müssen, um später mit dem entsprechenden Befehl auch den richtigen Kanal zu zünden!

Zunächst müssen die 16 Kabel mit den zugehörigen schwarzen Klemmen verbunden werden, was anhand des Layouts in Abbildung 6.8 auf Seite 71 erklärt werden soll. Kanal 1 wird vom Transistor Q2 gesteuert, Kanal 2 von Q3, Kanal 3 von Q4 und allgemein Kanal x-1 von Qx. Entsprechend ist das Kabel, welches an derjenigen Lötstelle angelötet ist, die durch die dicke blaue Linie unmittelbar mit dem mittleren Pin (Drain) von Q2 verbunden ist, an der Lötfahne der schwarzen Klemme ganz unten links – bezogen auf Abbildung 1.2b – anzubringen. Das Kabel an der Drain von Q3 (Zick-Zack-Anordnung der Transistoren beachten!) wird mit der schwarzen Klemme daneben verbunden, das an Q4 mit der dritten und das an Q5 schließlich mit der letzten Klemme in der Reihe, die unmittelbar neben der Viererreihe für die Kanal-LEDs liegt. Man sollte sich nicht dadurch verwirren lassen, dass aufgrund des umgedrehten Deckels alles seitensverkehrt ist, man die Klemmen also beim Löten von rechts nach links belegt. Analog zum bisherigen Vorgehen verfährt man in der Reihe darüber und den beiden anderen Reihen.

Nun werden mit Heiß- oder Sekundenkleber die Kanal-LEDs in die richtige Position gesteckt.

Für Kanal n ist dabei immer auch LEDn zuständig, für Kanal 1 also LED1, die über R21 mit der Drain von Q2 verbunden ist, für Kanal 2 LED2, usw. Sinnvollerweise ist LED1 in das Loch zu kleben, welches in der untersten Reihe direkt neben den Klemmen liegt, LED8 dementsprechend in der zweituntersten Reihe ganz außen usw.

Danach werden die vier Kabel am Leistungswiderstand jeweils mit einem der gespannten Drähte an den roten Klemmen verbunden. Hierbei spielt die Zuordnung (welches Kabel an welchen Draht?) keine Rolle. Damit sind sämtliche Kabel im inneren der Box nun verlötet und die Arbeit nähert sich langsam dem Ende!

Nun die SMA-Buchse fest am Gehäuse anschrauben, die Antenne vorerst dort befestigen und das andere Kabelende mit der Buchse am RFM-Adapter verbinden. Unter möglichst geringer Torsion sollte dann der RFM-Adapter mit dem aufgelöten Funkmodul in die Buchsenleisten auf der Platine gesteckt werden.

Wenn der Bootloader bereits auf den Controller geflasht wurde, kann man jetzt die beiden Seitenwände einstecken und die Box zuschrauben (wenn man sich sicher ist, dass sie funktioniert...). Ansonsten muss die Box noch offen bleiben.

Den männlichen Teil des Steckerpaars mit den aus der Box kommenden Batteriekabeln verlöten (Schrumpfschlauch nicht vergessen!), den weiblichen über zwei Kabel an der Batterie, wobei jeweils auf die korrekte Polung zu achten ist<sup>11</sup>.

Für die erste Inbetriebnahme sollte, wie zu Beginn des Kapitels empfohlen, idealerweise eine Laborspannungsquelle, ein kurzschlussfestes Steckernetzteil oder aber eine träge 3 A-Sicherung in der Zuleitung verwendet werden. Nun die Box über den Netzschalter einschalten und, falls noch nicht geschehen, den Bootloader via ISP flashen. Nun kann die Box zugeschraubt und die Firmware eingespielt werden.

### 12.4.3 Koffer

Als Aufbewahrungsort für die Zündboxen tut ein robuster Aluminiumkoffer wertvolle Dienste, um die Zündboxen vor Wettereinflüssen und Feuerwerksniederschlag zu schützen. Als gutes Pendant zum Plastikgehäuse bietet sich der in der Materialiste in Tabelle 10.4 aufgeführte Alu-Koffer 545 von Bilora an, der gerade ausreichend Platz für eine Zündbox und den zugehörigen Blei-Gel-Akku für die Stromversorgung bietet.

#### Schaumstoffeinlage

Der Koffer besitzt eine gewürfelte Schaumstoffeinlage mit einem  $18 \times 14$ -Raster, die gemäß Abbildung 12.8 entsprechend angepasst werden kann, um auch beim Transport einen festen Stand von Zündbox und Akku im Koffer zu ermöglichen.

Hierfür müssen für die Box  $13 \times 7$  sowie für den Akku  $6 \times 3$  Würfel händisch oder mit Hilfe eines Teppichmessers an den gezeigten Stellen gelöst werden. Zudem empfiehlt es sich noch, zwei Würfel

---

<sup>11</sup>Bei Deans-T-Steckern wird der obere Balken des T mit + verbunden, bei Tamiya-Steckern das eckige Profil

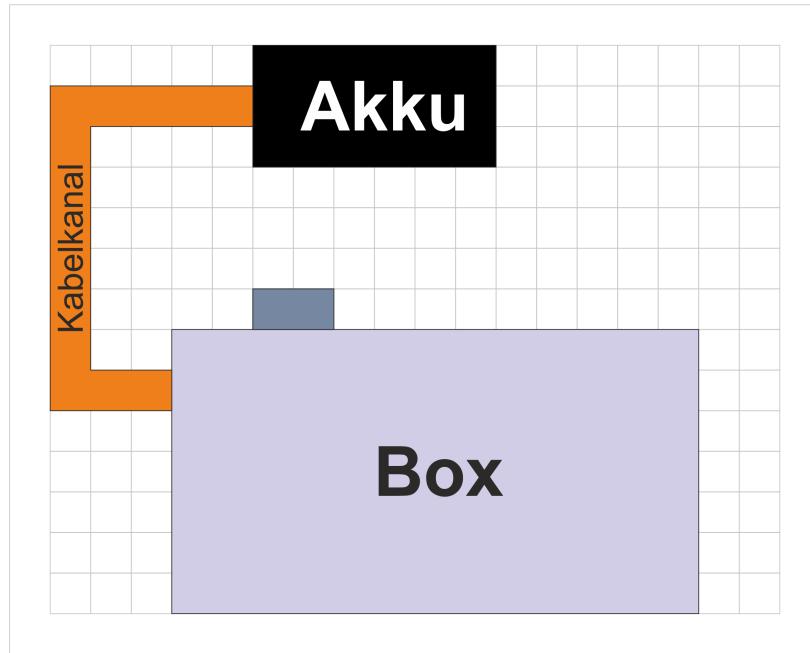


Abbildung 12.8: Ausschnitt aus der Schaumstoffeinlage für Zündbox und Akku

an der seriellen Schnittstelle für ein leichteres Herausnehmen der Box und den eingezeichneten Kabelkanal herauszutrennen. Um ein Herauslösen weiterer Würfel zu verhindern, kann mittels Sekunden- oder Heißkleber in den Kreuzungen der vorgestanzten Linien eine festere Naht hergestellt werden.

### Antennenhalterung

Etwas Arbeit ist nötig, um die Antenne in eine geeignete Position zu bringen, um bei geschlossenem Koffer die Zündkommandos noch sicher und zuverlässig empfangen zu können. Wenn die Antenne nicht außerhalb des Gehäuses angebracht ist, können ankommende Signale nicht bzw. nur äußerst stark gedämpft hinein- und abgehende so gut wie nicht hinausgelangen, weil der Alukoffer als Faradayscher Käfig wirkt.

Um die Antenne vom SMA-Anschluss an der Zündbox nach außen zu führen, wird ein zweiteiliger Adapter, der in Abbildung 12.9 am Koffer montiert gezeigt ist, verwendet. So wird einerseits das Kabel geschützt, andererseits die Kabelführung stabilisiert. Die Halterung besteht aus zwei ineinander gesteckten Teilen:

- MDF-Platte: 85 mm × 35 mm × 3 mm
- PVC-Rohr, basierend auf einer Gardena-Micro-Drip-Verlängerung: Ø7,5 mm × 115 mm

In die MDF-Platte wird 10 mm vom oberen Rand entfernt mehr oder weniger mittig ein 5 mm-Loch

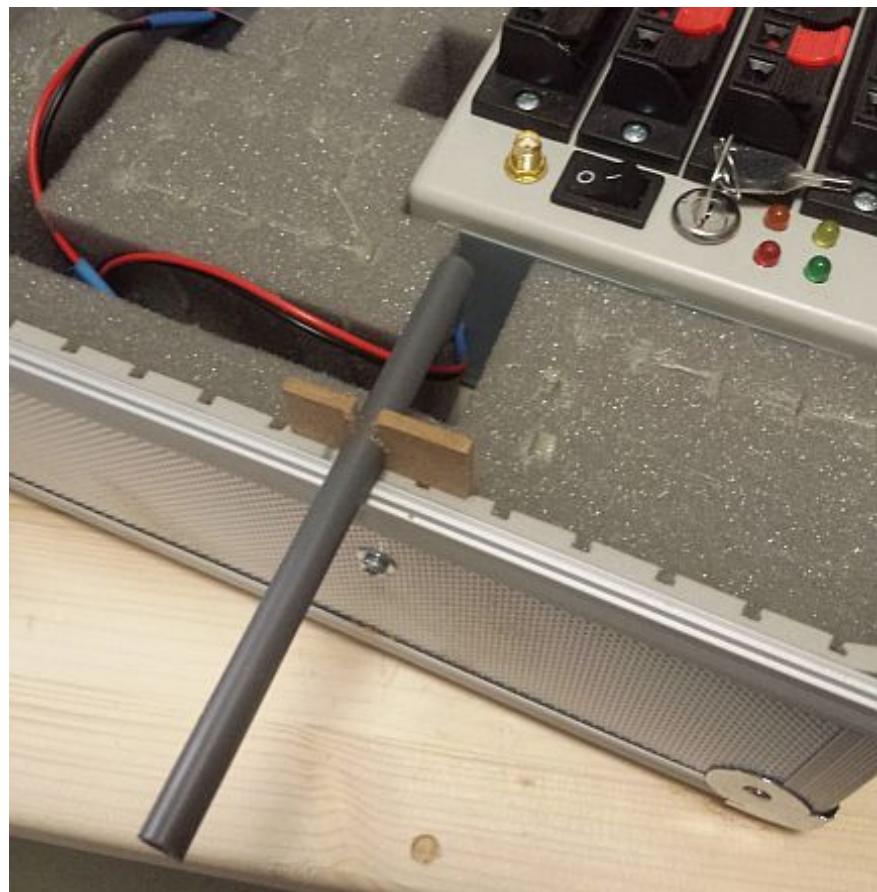


Abbildung 12.9: Adapter am Koffer montiert



(a) Einzelteile



(b) Zusammen

Abbildung 12.10: Antennenadapter

gebohrt und ein Kanal senkrecht nach oben zum Rand eingebracht. Beim PVC-Rohr bringt man 7 cm vom Rand entfernt auf einer Länge von 3 mm zwei Einschnitte ein, um das Rohr an dieser Stelle auf die Breite des Kanals zu bringen. Die beiden Einzelteile sind in Abbildung 12.10a, der zusammengesteckte Adapter in Abbildung 12.10b abgebildet. Der Adapter wird vor dem Montieren der MDF-Platte am Koffer so ausgerichtet, dass die Rohröffnung genau auf die SMA-Buchse an der Zündbox zeigt.

Nun muss das Antennenkabel durch das Rohr geführt werden, so dass der gewinkelte männliche SMA-Stecker an dem Ende des Rohrs liegt, welches näher an den Einschnitten liegt. Wird ein bereits fertig konfektioniertes Antennenkabel, z. B. das in der Materialliste aufgeführte, verwendet, muss das PVC-Rohr auf kompletter Länge aufgetrennt werden, um das Kabel einführen zu können. Konfektioniert man das Kabel selbst, kann die Buchse auf der Antennenseite nach dem Einfädeln angebracht werden, womit man sich das Auf trennen des Rohrs spart. Ist das Kabel durch das Rohr gefädelt und sind Winkelstecker und Buchse angebracht, sollte der gewinkelte Stecker kurz an der Box angeschraubt und der Adapter zusammengesteckt werden, damit man die nötige Länge für den Schrumpfschlauch zwischen MDF-Platte und Winkelstecker ausmessen kann. Dieser wird dann, nach Abschrauben des Winkelsteckers und Herausnehmen des Rohrs vom „Buchsenende“ her aufgeschoben und erhitzt.

Im nächsten Schritt muss das Rohr außerhalb des Koffers gebogen werden, damit die Antenne später senkrecht nach oben zeigt. Hierzu sollte das PVC-Rohr mit einem Heißluftföhn erwärmt und in größerem Radius (Koaxialkabel sollten nie geknickt werden!) gebogen werden.

Zum Schluss wird die Antenne an die Buchse angeschraubt und die Verschraubung mit Schrumpfschlauch überdeckt. Die fertige Antenne, die außerhalb der Einsatzzeiten der Zündbox abgeschraubt im Koffer aufbewahrt werden kann, ist in Abbildung 12.11 dargestellt.

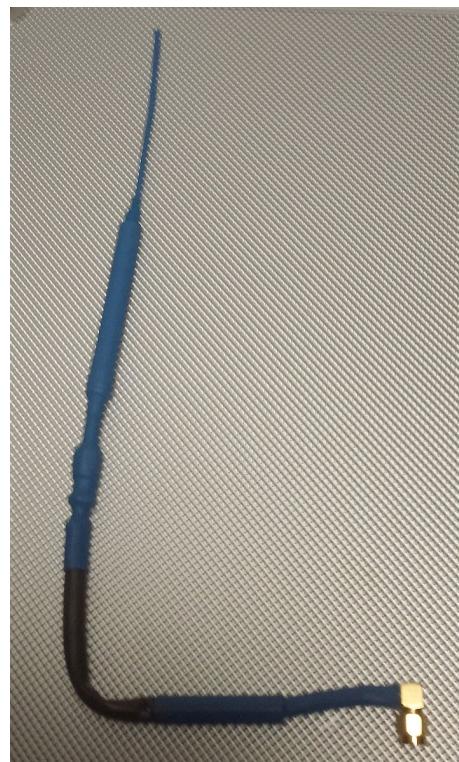


Abbildung 12.11: Einsatzbereite Antenne mit Zuleitung

#### 12.4.4 Gehäuse des Raspberry-Pi-Transmitters

Über je einen der USB-Ports am Raspberry Pi sind der Touchscreen sowie die USB-Soundkarte anzuschließen, für die Displayfunktionalität des Touchscreens wird zusätzlich eine HDMI-Verbindung benötigt.

# 13 Aufspielen des Bootloaders

Bevor Firmwareupdates über die serielle Schnittstelle eingespielt werden können, muss zunächst ein Programm auf den Controller gespielt werden, dessen Aufgabe es ist, die eigentliche Firmware in den Speicher zu laden und zu starten. Dieses Programm ist der so genannte Bootloader, welcher beim Start des Devices für eine Sekunde überprüft, ob ein Firmwareupdate vorgenommen werden oder die *El Fueradore*-Firmware normal ausgeführt werden soll.

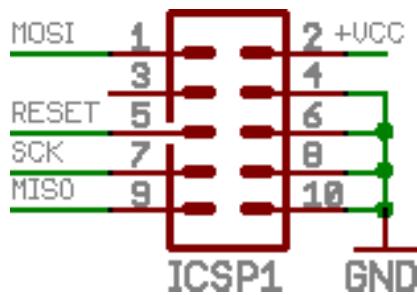


Abbildung 13.1: Pinbelegung des ISP-Platinensteckers: Ansicht von oben, Gehäuseaussparung an Pin 5. Quelle: [mikrocontroller.net](http://mikrocontroller.net)

Um den Bootloader auf den Controller zu brennen und einige Grundeinstellungen des Controllers, die so genannten Fuses, welche neben den Einstellungen, welche die Verwendung eines Bootloaders ermöglichen, auch Funktionen wie die Brown-Out-Detektion oder die Taktquelle regeln, wird ein spezielles Programmiergerät zur In-System-Programmierung (ISP) benötigt, welches den Controller in den Resetzustand versetzt und anschließend den Bootloader über die SPI-Schnittstelle an eine festgelegte Stelle im Flash-Speicher des Controllers schreibt.

Auf der Platine jedes Devices ist für ISP ein zehnpoliger zweireihiger Wannenstecker vorgesehen, an den gängige Programmiergeräte wie der weit verbreitete *AVRISP mkII* angeschlossen werden. Seine Pinbelegung ist in Abbildung 13.1 gezeigt. Nach einmaligem Flashen des Bootloaders und EEPROMs wird die ISP-Schnittstelle nicht wieder benötigt, alle weiteren Änderungen können über die serielle Schnittstelle und den Bootloader vorgenommen werden.

## 13.1 Verwendung des AVRISP mkII

Zum Brennen des Bootloaders gibt es ein Kommandozeilentool namens *bldflsh.exe* für den *AVRISP mkII*, welches auf *AVRDUEDE* basiert. Um einen mit kompatiblen USB-Treiber für den Programmier-

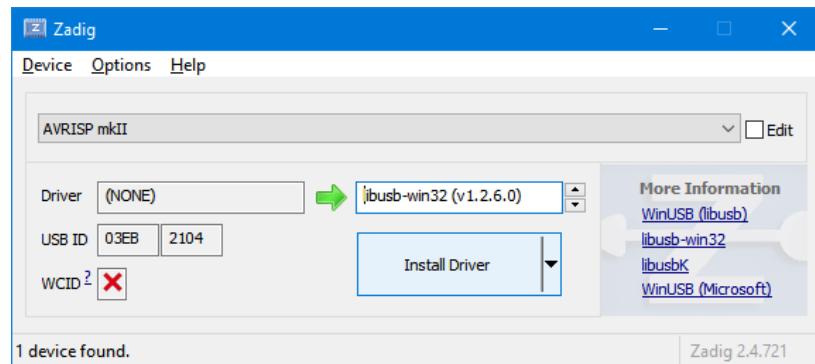


Abbildung 13.2: Einstellungen in Zadig zur Erzeugung und Installation eines USB-Treibers für den Programmierer

adapter zu installieren, gibt es unter Windows das Programm Zadig<sup>12</sup>, in dem vor dem Klick auf „Install Driver“ die Einstellungen gemäß Abbildung 13.2 zu treffen sind.

Dem Tool muss als Parameter die iHex-Datei des controllertyp- und frequenzspezifischen Bootloaders übergeben werden.

Für einen ATmega328P mit einer Taktfrequenz von 9,8304 MHz lautet das Kommando:

```
btlflash.exe bootload_m328p_9830400.hex
```

Die für *El Fueradoro* benötigten Einstellungen für Fuses und die Datenübertragung werden auf diese Weise automatisch angepasst. Ebenfalls übertragen wird beim Flashen des Bootloaders eine Standardversion des EEPROMS, so dass die Devices standardmäßig 1 als Unique- und Slave-ID zugewiesen bekommen (Transmitter rekonfigurieren sich dann automatisch beim ersten Start der Firmware).

## 13.2 Verwendung eines anderen Programmieradapters

Selbstverständlich ist das Aufspielen des Bootloaders auch mit anderen Programmern möglich.

Wichtig für die ordnungsgemäße Funktion des Bootloaders sowie später der Firmware ist neben einer fehlerfreien Programmierung auch das korrekte Setzen der Fuse-Bits, welches bei Einsatz eines alternativen Programmieradapters manuell vorgenommen werden muss.

Für *El Fueradoro* müssen die Fusebits beim ATmega328P gemäß Tabelle 13.1 gesetzt werden.

<sup>12</sup>Zadig-Download: <https://zadig.akeo.ie/>

Fuse	Wert	Bedeutung
Low Fuse	0xF7	Kein Taktteiler, kein Clock-Output, Ext. Full Swing Crystal als Taktquelle
High Fuse	0xD6	Reset-Pin nicht als I/O-Pin, kein Debug-Wire, SPI-Download erlaubt, Watchdog aus, EEPROM nicht löschen, Bootbereich = 256 Wörter, Boot-Reset-Vektor aktiviert (=nach Reset Bootloader starten)
Extended Fuse	0x05/ 0xFD <sup>13</sup>	Brown-Out bei Versorgungsspannung unter 2,7 V

Tabelle 13.1: Fuse-Einstellungen beim ATmega328P

Wird beim Flashen des Bootloaders das Standard-EEPROM-Image nicht mitübertragen, die herstellerseitige Voreinstellung des EEPROMs also nicht verändert, werden Zündboxen – nach dem Programmieren der „echten“ Firmware – zunächst „E“ bzw. „e“ für „Error“ als Unique- bzw. Slave-ID melden, da an den Speicherstellen für IDs und Prüfsummen nicht zueinander passende Werte stehen. Die Zündbox muss dann einmalig kabelgebunden über die lokale Konfiguration auf gültige Werte eingestellt werden.

Wird das Tool *AVRDUDE* verwendet, müssen der Programmername als Parameter hinter „-c“ und die Verbindung zum Computer – mögliche Werte sind unter anderem *usb* oder *comXY*, wobei XY eine Zahl bezeichnet – hinter „-P“ angegeben und an der entsprechenden Stelle in folgenden Befehl eingefügt werden:

```
avrduude -p m328p -c PROGRAMMER -P VERBINDUNG -u
-Uflash:w:bootload_m328p_9830400.hex:a -U eeprom:w:std_eep.eep:a -Ulfuse:w:0xf7:m
-Uhfuse:w:0xd6:m -Uefuse:w:0xfd:m -v
```

<sup>13</sup>Bei der Extended Fuse werden nur die unteren drei Bit verwendet, die oberen fünf sind nicht in Gebrauch und können daher beliebig jeweils mit 1 oder 0 beschrieben und gelesen werden. Werden die nicht-relevanten Bits mit 0 beschrieben, ergibt sich der Wert 0x05, bei Beschreibung mit 1 der Wert 0xFD. AVRDUDE ist leider nicht imstande, die oberen fünf Bits zu ignorieren und meldet daher einen Verifizierungsfehler, falls die Bits mit 0 geschrieben und mit 1 gelesen bzw. mit 1 geschrieben und mit 0 gelesen werden.



# 14 Micro-SD-Karte für raspEasyFire vorbereiten

## 14.1 Image herunterladen und übertragen

Aufgrund der Vielzahl an notwendigen Einstellungen und Zusatzpaketen wird für den schnellen Einstieg ein fertiges *raspEasyFire*-Image auf Basis des Betriebssystems Raspbian zur Verfügung gestellt, welches mit einem geeigneten Tool auf eine Micro-SD-Karte geschrieben werden kann. Das Image besitzt eine Größe von etwa 5 GB und kann in seiner jeweils aktuellsten Version heruntergeladen werden unter:

<http://raspeeasyfire.de/raspEasyFire.img>

Um diese Datei nun so auf eine Micro-SD-Karte (Mindestkapazität 8 GB) übertragen zu können, dass sie von *raspEasyFire* genutzt werden kann, bietet sich unter Windows das kostenlose Tool **Win32 Disk Imager**<sup>14</sup> an. Hier muss nun einfach die soeben heruntergeladene Imagedatei geladen, der korrekte Laufwerksbuchstaber der Micro-SD-Karte ausgewählt und anschließend der Schreibvorgang gestartet werden, der einige Minuten in Anspruch nimmt. Unter Linux kann mit dem Kommandozeilentool DiskDump (dd) gearbeitet werden.

**Wichtig:** Soll unter Windows eine bereits mit einem Image beschriebene SD-Karte neu beschrieben werden, sind zunächst folgende Schritte mit dem Windows-Boardtool DISKPART notwendig, um die Karte erneut nutzen zu können:

1. Einlegen der SD-Karte in den Windows-Rechner
2. Starten der Kommandozeile durch Drücken der Windows-Taste zusammen mit „R“, Eingabe von „cmd“ und Klick auf OK.
3. Aufrufen des Tools „diskpart“
4. In der nun gestarteten Eingabemaske, die mit „DISKPART>“ vor dem Cursor beginnt, lässt man die verfügbaren Datenträger (Festplatten und SD-Karten) mit dem Befehl „list disk“ anzeigen.

---

<sup>14</sup>Setupdatei zur Installation herunterzuladen unter: <https://sourceforge.net/projects/win32diskimager/files/latest/download>

5. Anhand der Größenangabe in der dritten Spalte identifiziert man nun die SD-Karte und merkt sich die zugehörige Nummer in der ersten Spalte hinter „Datenträger“, z.B. die „2“.
6. Die im vorherigen Schritt gemerkte Datenträger-Nummer wird nun über den Befehl „select disk NUMMER“, also im Beispiel durch „select disk 2“ ausgewählt und sollte vom Programm bestätigt werden.
7. Durch den Befehl „clean“ bereinigt man den Datenträger vollständig.
8. Nun muss durch „create partition primary“ eine neue Partition erstellt werden.
9. Diese Partition wird über „select partition 1“ ausgewählt.
10. Durch „active“ bestätigen.
11. Die Partition mit format fs=fat32 quick formatieren.
12. Durch „assign“ bestätigen.
13. Diskpart durch „exit“ beenden.

Ers jetzt kann mit dem **Win32 Disk Imager** ein neues Image korrekt geschrieben werden.

Hintergrund ist, dass das Image zwei Partitionen (Boot- und Datenpartition) enthält, wobei für Windows selbst nur die Bootpartition verarbeitbar ist, da es sich bei der Datenpartition um eine Linux-Partition im ext4-Format handelt, die Windows nicht erkennen kann. Sämtliche Aktionen aus dem Windows-Explorer würden sich ohne die DISKPART-Schritte daher nur auf die Bootpartition beziehen und die Datenpartition unangetastet lassen.

## 14.2 WLAN-Daten einstellen

Damit *raspEasyFire* problemlos via SSH-Verbindung über WLAN von einem Windows-Rechner, über den erfahrungsgemäß die meisten Anwender verfügen, konfiguriert werden kann, können vor dem Einlegen der Micro-SD-Karte in den Raspberry Pi noch die Zugangsdaten des eigenen Drahtlosnetzwerks auf der Karte hinterlegt werden.

Nach erfolgreichem Übertragen des Images sollte Windows die Karte unter der Bezeichnung „boot“ erkennen<sup>15</sup>. Im Hauptverzeichnis kann nun eine Textdatei namens „wifidata.txt“ angelegt werden, die aus zwei Zeilen besteht. Die erste Zeile enthält die SSID des Drahtlosnetzwerks, die zweite Zeile den Schlüssel. *raspEasyFire* sucht beim Hochfahren nach dieser Datei und wird ein WLAN mit den

---

<sup>15</sup>In Wahrheit besteht das System neben der Boot-Partition noch aus einer weiteren Partition, die aufgrund ihres Formats für Windows aber nicht erkennbar ist.

entsprechenden Zugangsdaten an der korrekten Stelle im System eintragen, so dass eine automatische Drahtlosverbindung hergestellt werden kann.

Sollte *raspEasyFire* nach dem Hochfahren nicht im Netzwerk auftauchen, liegt dies möglicherweise an vorhandenen MAC-Filtern und Zugangsbeschränkungen im Router, die zumindest bis zur ersten erfolgreichen Verbindung deaktiviert werden müssen.

Nun kann die Speicherkarte in *raspEasyFire* eingeführt und verwendet werden.



# 15 Tipps und Tricks

## 15.1 5V-LCD an 3,3V

Zwar gibt es inzwischen auch LCDs, welche sich von Haus aus für eine Versorgung mit 3,3 V eignen, viele Displays jedoch sind noch für 5 V ausgelegt. Der interne Controller funktioniert ohne Probleme auch bei geringerer Spannung, Knackpunkte sind jedoch die Kontrastspannung für das LCD sowie die Hintergrundbeleuchtung.

Die Kontrastspannung wird zwischen Pin 2 und Pin 3 gemessen und ist verantwortlich für die Lesbarkeit der Schrift auf dem Display. Im „Normalfall“ – also bei Betrieb des Displays mit einer Versorgungsspannung von 5 V – wird Pin 3 auf 0 V gelegt, so dass sich eine Kontrastspannung von 5 V einstellt. Liegen an Pin 2 nur 3,3 V an, muss Pin 3 folglich mit einer negativen Spannung verbunden werden, um die nötige Kontrastspannung zu erreichen.

Als Quelle der negativen Spannung dient der RS232-Treiberbaustein, der am Pin V- eine Spannung von -5,5 V zur Verfügung stellt. Über einen Spannungsteiler – im Schaltplan auf Seite 64 von  $R_{12} = 3,3\text{ k}\Omega$  und  $R_{13} = 6,8\text{ k}\Omega$  gebildet – zwischen V- und GND wird daher Pin 3 des LCD auf -1,8 V gelegt.

Manche LCDs torpedieren diesen Versuch, indem Pin 3 relativ niederohmig mit GND verbunden wird. Man sollte also im abgeklemmten Zustand den Widerstand zwischen Pin 1 und Pin 3 messen und bei Bedarf den eventuell auf der LCD-Platine befindlichen Widerstand zwischen den beiden Pins auslöten.

Die Hintergrundbeleuchtung wird über die Pins 15 und 16 versorgt. Zwischen diesen Pins befindet sich in Reihe zu den LEDs in aller Regel noch ein Widerstand, welcher den Strom durch die LEDs begrenzt und für eine Spannung von 5 V zwischen den Pins ausgelegt ist. Bei 3,3 V zwischen Pin 15 & 16 erscheint das LCD daher möglicherweise zu dunkel, so dass man den Widerstand durch einen kleineren Wert ersetzen kann, um die Spannungsdifferenz auszugleichen.

Der ursprüngliche LED-Vorwärtsstrom sowie die Aufteilung der Spannung auf LED und Vorwiderstand können durch Anlegen von 5 V zwischen Pin 15 & 16, Spannungsmessung über dem Original-Widerstand und anschließende Division durch den Widerstandswert (korrekt ablesen oder messen) ermittelt werden. Da der Spannungsabfall über den LEDs sich nicht ändert und der Strom gleich bleiben soll, muss der Wert des neuen Widerstands so verkleinert werden, dass bei identischem Stromfluss über ihm 1,7 V weniger abfallen als am Original-Widerstand.

Ist z. B. original ein Widerstand von  $150\Omega$  verbaut, über dem eine Spannung von 2 V anliegt (hieraus resultiert eine LED-Vorwärtsspannung von 3 V), ergibt sich ein LED-Vorwärtsstrom von 13 mA. Dementsprechend wäre bei einem Spannungsabfall von nur noch 0,3 V für den gleichen Strom ein Widerstand von  $22\Omega$  einzusetzen.

## 15.2 Antennenbau

Antennen für die verwendete Übertragungsfrequenz von 868 MHz gibt es in großer Auswahl zu kaufen, eine einfache, omnidirektionale Antenne, welche ein sehr gutes Stehwellenverhältnis von <1,3:1 erzielt, kann aber auch relativ schnell selbst gebaut werden. Auf möglichst exakte Einhaltung der Abmessungen ist dabei zu achten:

- Koaxialkabel RG316
- SMA-Steckverbinder (üblicherweise männliche Ausführung)
- Kupfer- oder Messingrörchen mit 8 mm Durchmesser und 66,5 mm Länge
- Distanzhülse aus Kunststoff mit 5 mm Länge, 7 mm Außendurchmesser und 3,6 mm Innen-durchmesser
- Schrumpfschläuche mit 1,2 mm, 2,4 mm, 4,8 mm und 9,5 mm Durchmesser vor dem Schrumpfen

Die Antenne wird als Sperrtopfantenne bezeichnet und besitzt den in Abbildung 15.1 gezeigten Aufbau. Der oberste Teil ist 88 mm lang und besteht aus dem Innenleiter des Koaxialkabels mit dem ihn umgebenden Dielektrikum. Im mittleren Teil befindet sich ein 66,5 mm langes Röhrchen, welches an seinem oberen Ende mit möglichst kurzer Verbindung an das Schirmgeflecht des Koaxialkabels angelötet wird. Anschließend folgt eine beliebige Länge Koaxialkabel, am Ende schließlich der Steckverbinder zum Anschluss an das Funkmodul.

Der Aufbau der Antenne erfolgt folgendermaßen:

1. Von der Rolle RG316 ein Stück der Länge abschneiden, welche später der Gesamtlänge von Antennenspitze bis zum Anschluss an das Funkmodul bzw. einen Adapterstecker entspricht
2. Entfernung des Kunststoffmantels auf einer Länge von 88 mm
3. Entfernung des nun freiliegenden Schirmgeflechts auf einer Länge von 84 mm, so dass noch 4 mm des Schirmgeflechts verbleiben.
4. Auftrennen und Verdrillen des Schirmgeflechts



Abbildung 15.1: Antenne ohne Schrumpfschlauch (links) und komplett fertig (rechts)

5. Verdicken des Koaxialkabels mit einem 10 mm langen Stück Schrumpfschlauch, dessen Mitte 66,5 mm vom oberen Ende der Ummantelung entfernt sein sollte.
6. Aufschieben der Distanzhülse in die Mitte des soeben verstärkten Teils. Gegen Abrutschen nach unten ggf. mit weiterem Schrumpfschlauch unterhalb der Hülse sichern.
7. Großzügiges Vorverzinnen des Röhrchens an einer Stelle der Innenwand
8. Überstülpen des Röhrchens und Anlöten des verdrillten Schirmgeflechts an der Innenwand
9. Gesamte Konstruktion mit Schrumpfschlauch stabilisieren (nach jedem Schritt schrumpfen!):
  - a) Ein 88 mm langes Stück Schrumpfschlauch 1,2 über den obersten Teil der Antenne
  - b) Ein 90 mm langes Stück Schrumpfschlauch 2,4 über den obersten Teil der Antenne, unmittelbar nach dem Schrumpfen die noch heißen oben überstehenden 2 mm Schlauch durch Zusammendrücken verschmelzen
  - c) Ein 2,5 mm langes Stück Schrumpfschlauch 9,5 über den aus dem Röhrchen herausstehenden Teil der Distanzhülse

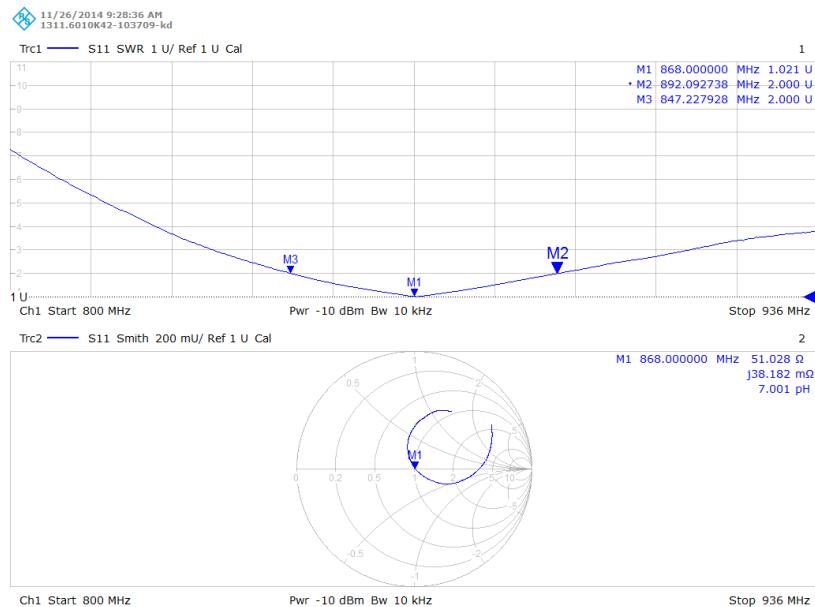


Abbildung 15.2: VSWR-Messung der gefertigten Antenne am Netzwerkanalysator

- d) Ein 80 mm langes Stück Schrumpfschlauch 9,5 über Röhrchen und Distanzhülse, so dass auf beiden Seiten etwa 5 mm überstehen
  - e) Mit einem 10 mm langen Stück Schrumpfschlauch 4,8 den Übergang zwischen oberem Antennenteil und Röhrchen versiegeln
10. Falls nötig: Kabel in Endposition einfädeln, bevor SMA-Steckverbinder am intakten Ende angebracht wird

Um den SMA-Steckverbinder anzubringen, müssen vom intakten Ende aus gemessen zunächst 12 mm des Mantels entfernt werden, anschließend 8 mm des Schirmgeflechts und zuletzt auch 4 mm des Dielektrikums. Nun werden Schrumpfschlauch und Crimpröhrchen auf das Kabel geschoben, anschließend der kleine Stecker am Innenleiter angelötet. Das Schirmgeflecht wird aufgefächert, das Gehäuse aufs Kabel geschoben und das Crimpröhrchen aufgesteckt, vercrimpft und Schrumpfschlauch darüber angebracht. Wenn gewünscht kann – wie in 15.1 rechts zu sehen – Schrumpfschlauch 9,5 als Witterungsschutz über dem gesamten Steckverbinder angebracht werden.

Die in Abbildung 15.2 dargestellte Messung am Netzwerkanalysator zeigt eine sehr gute Anpassung dieser Antenne an  $50 \Omega$  im Bereich um 868 MHz.

## 15.3 Kompilieren der Firmware

Falls keine der im Repository zur Verfügung gestellten Hex-Files verwendet werden soll, besteht auch die Möglichkeit, seine eigene Firmware aus den Quellcodedateien zu kompilieren. Hierfür wird der die AVR-GCC-Toolchain benötigt, welche es für Windows z. B. als fertiges Paket **WinAVR**<sup>16</sup> gibt.

**WinAVR** entspricht leider schon länger nicht mehr dem aktuellen Stand des GCC-Compilers, man kann entweder bei Atmel<sup>17</sup> oder bei im Sourceforge-Repository<sup>18</sup> von Georg-Johann Lay aktuellere Versionen ziehen und diese über die alten **WinAVR**-Verzeichnisse kopieren.

Anschließend ist darauf zu achten, dass das Unterverzeichnis „bin“ in der PATH-Variable hinterlegt ist, so dass man die dort befindlichen Anwendungen aus jedem anderen Ordner unmittelbar aufrufen kann.

Der Herstellungsprozess gliedert sich in folgende Schritte:

1. Kompilieren der Quellcodedateien (.c) in noch nicht gelinkte Dateien (.o)
2. Linken zu einer Gesamtdatei (.elf)
3. Umwandeln in eine Hexdatei (.hex), welche mit dem Bootloader über die serielle Schnittstelle auf den Controller geschrieben werden kann

Der gesamte Ablauf kann transparent in der Datei „build\_hexfiles.bat“ im Verzeichnis „Hexfiles“ nachvollzogen werden.

## 15.4 Umrechnung zwischen Watt und dBm

Die Umrechnung zwischen den Einheiten Watt (linear) und dBm (logarithmisch) erfolgt nach folgenden Formeln:

$$\begin{aligned} \text{dBm} \rightarrow \text{W} : \quad P_W &= 10^{\left(\frac{P_{\text{dBm}}}{10} - 3\right)} \\ \text{W} \rightarrow \text{dBm} : \quad P_{\text{dBm}} &= 10 \cdot \log_{10} \left( \frac{1000 \cdot P_W}{1 \text{W}} \right) \end{aligned}$$

Anhand der Formeln wird klar, dass die maximale Sendeleistung des RFM69CW 1250-mal höher als die Minimalleistung liegt: –18 dBm entsprechen 16 µW, 13 dBm hingegen 20 mW.

---

<sup>16</sup>Download WinAVR

<sup>17</sup>Download Atmel-Toolchain (kostenlose Registrierung notwendig)

<sup>18</sup>Downloadlink: Sourceforge

Addition und Subtraktion in der logarithmischen Einheit sind gleichbedeutend mit Multiplikation bzw. Division in der linearen. Erhöht (verringert) sich die Leistung um 3 dB, verdoppelt (halbiert) sie sich im linearen Maßstab. Addition bzw. Subtraktion um 10 dB bedeuten eine Multiplikation mit bzw. Division durch 10.

# Abbildungsverzeichnis

1.1	Systemübersicht <i>El Fueradoro</i>	12
1.2	Die Devices von <i>El Fueradoro</i>	13
1.3	LCD während der Show	17
1.4	Schalter und serielle Schnittstelle an der Zündbox	17
2.1	Eintrag des USB-RS232-Adapters im Gerätemanager	20
2.2	Einstellungen für <b>Puttytel</b>	21
2.3	Einstellungen in <i>Pyro Ignition Control</i>	23
3.1	Ablauf des Konfigurationsprogramms bei Verbindung mit einer Zündbox	27
3.2	Beispiel einer ferngesteuerten ID-Zuweisung	28
3.3	Systemübersicht	30
3.4	Auslesen der gesetzten Sendeleistung	32
3.5	Setzen der Sendeleistung mit anschließendem Auslesen	32
3.6	Auslesen und Neusetzen des AES-Schlüssels	33
3.7	Startbildschirm der GUI	34
3.8	Sendebildschirm der GUI	34
3.9	Netzwerkanzeige der GUI	35
3.10	Lokale ID-Konfiguration	36
3.11	Remote-ID-Einstellung	37
3.12	Benutzeroberfläche <b>UpdateLoader</b>	38
4.1	Aufgebautes System <i>raspEasyFire</i>	40
4.2	Startbildschirm nach dem Aufruf von <i>raspEasyFire</i>	44
4.3	Auswahl des Zündplans	45
4.4	Anzeige der gefundenen Boxen	47
4.5	Ändern der IDs	49
4.6	Widerstandsmessung	50
4.7	Bildschirme während der Show	53
6.1	Schaltplan des Transmitters	64
6.2	Schaltplan der Zündbox	65
6.3	Schaltplan der Zündbox (2. Generation)	66
6.4	Schaltplan der Zündbox (3. Generation)	67
6.5	Schaltplan der Zündbox (3. Generation) mit 32 Kanälen	68
6.6	Schaltplan des Raspberry-Pi-Adapters	69
6.7	Layout des Transmitters (keine Originalgröße!)	70

6.8	Layout der Zündbox (keine Originalgröße!) . . . . .	71
6.9	Layout der Zündbox (2. Generation, keine Originalgröße!) . . . . .	72
6.10	Layout der Zündbox (3. Generation, keine Originalgröße!) . . . . .	73
6.11	Layout der Zündbox (3. Generation mit 32 Kanälen, keine Originalgröße!) . . . . .	74
6.12	Layout des Raspberry-Pi-Adapters (keine Originalgröße) . . . . .	75
8.1	Pinbelegung des Mikrocontrollers bei Transmitter und Zündboxen . . . . .	79
8.2	Belegung Verbindungsstiftleiste Pi-Aufsteckplatine . . . . .	81
8.3	Pinbelegung der Aufsteckplatine von oben gesehen . . . . .	81
11.1	Ober- und Unterseite des Transmitters für Toner-Transfer-Verfahren/Belichtung . . . . .	100
11.2	Ober- und Unterseite der Zündbox für Toner-Transfer-Verfahren/Belichtung . . . . .	101
11.3	Adapterplatine für Funkmodule (8 Stück) für Toner-Transfer-Verfahren/Belichtung . . . . .	102
12.1	Platinen für Transmitter (oben) und Zündbox (unten) . . . . .	105
12.2	Zündboxplatine (1. Generation) vor dem Einlöten der Kabel und MOSFETs . . . . .	106
12.3	Raspberry-Pi-Erweiterungsplatine mit aufgestecktem Zündmodul und Antenne . . . . .	107
12.4	Fertig aufgebauter Funkmoduladapter . . . . .	108
12.5	Abmessungen (in mm) des LCD . . . . .	109
12.6	Bohrscharblone für Zündboxoberseite . . . . .	111
12.7	Innenansicht des Zündboxdeckels . . . . .	113
12.8	Ausschnitt aus der Schaumstoffeinlage für Zündbox und Akku . . . . .	115
12.9	Adapter am Koffer montiert . . . . .	116
12.10	Antennenadapter . . . . .	117
12.11	Einsatzbereite Antenne mit Zuleitung . . . . .	118
13.1	Pinbelegung des ISP-Platinensteckers: Ansicht von oben, Gehäuseaussparung an Pin 5. Quelle: <a href="http://mikrocontroller.net">mikrocontroller.net</a> . . . . .	119
13.2	Einstellungen in Zadig zur Erzeugung und Installation eines USB-Treibers für den Programmer . . . . .	120
15.1	Antenne ohne Schrumpfschlauch (links) und komplett fertig (rechts) . . . . .	129
15.2	VSWR-Messung der gefertigten Antenne am Netzwerkanalysator . . . . .	130

# Tabellenverzeichnis

1.1	Farben und Funktionen der Status-LEDs . . . . .	15
2.1	Konfiguration der seriellen Schnittstelle . . . . .	20
3.1	Kommandos zur Konfiguration über die serielle Schnittstelle . . . . .	26
3.2	Umrechnung Hexadezimal-, Dezimal- und Binärwerte . . . . .	31
3.3	Struktur des RFM69-Befehls . . . . .	31
3.4	Verfügbare Firmware-Versionen . . . . .	37
5.1	Widerstandswerte für Zündkreise bei gegebenem Strom . . . . .	56
5.2	Spezifischer Widerstand verschiedener Leitermaterialien . . . . .	57
9.1	Quellcodedateien und ihre Funktionen . . . . .	83
10.1	Materialliste für den Transmitter . . . . .	88
10.2	Materialliste für den Raspberry-Pi-Transmitter . . . . .	89
10.3	Materialliste für die Zündbox (1. Generation) . . . . .	91
10.4	Materialliste für die Zündbox (2. Generation) . . . . .	93
10.5	Materialliste für die Zündbox (3. Generation) . . . . .	95
10.6	Materialliste für die Zündbox (3. Generation) mit 32 Kanälen . . . . .	97
12.1	Übersicht über benötigte Kabelverbindungen . . . . .	104
13.1	Fuse-Einstellungen beim ATmega328P . . . . .	121



# Danksagung

Vielen Dank an alle, die in irgendeiner Form zur Entwicklung und Verbesserung von *El Fueradoro* beigetragen haben!

Ein ganz spezielles Dankeschön an:

- Marc Weissmann, der mit ursprünglich seinem, mittlerweile unserem gemeinsamen Projekt **FIRE – Feuerwerk im Röthelheimpark Erlangen** den Ansporn zu dieser Entwicklung gegeben und ihr von Beginn an trotz einiger mittlerweile behobener „Kinderkrankheiten“ immer voll vertraut hat. Videos unserer bisherigen mit *El Fueradoro* geschossenen Feuerwerke:
  - \* [Silvesterfeuerwerk 2014/15](#)
  - \* [Silvesterfeuerwerk 2015/16](#)
  - \* [Silvesterfeuerwerk 2016/17](#)
  - \* [Hochzeitsfeuerwerk April 2017](#)
  - \* [Hochzeitsfeuerwerk August 2017](#)
  - \* [Silvesterfeuerwerk 2017/18](#)
- Das gesamte private Umfeld, das die Arbeit am Projekt stets mit Fassung getragen und mit Interesse verfolgt hat.
- Jens Nachtigal, den Hauptprogrammierer und Anstoßgeber zur Entwicklung des Raspberry-Pi-Transmitters.
- Michael Czifra (pyro-michel) für die Unterstützung beim Layout der 32-Kanal-Version.
- Die „Nachbau-Pioniere“ Matthias Pee und Cedrik Sikora für Ideen und wertvolle Hinweise.
- **mikrocontroller.net** für Hilfe in Sachen Hardware und Programmierung.
- **feuerwerk-forum.de** für Erklärungen zum Thema elektrische Zündung und natürlich die Zündsoftware *Pyro Ignition Control* von User *pyrobla*.

- **Saleae** für den grandiosen Logic 8, der wertvolle Dienste zur Erschließung des **RFM69** geleistet hat.
- Der **Brauerei Loscher** für Club Mate, den Bastler-Treibstoff.
- Die vielen tollen Open-Source- und Freeware-Lösungen, welche die Softwareentwicklung erst möglich gemacht haben:
  - Die GCC-Community für die **AVR-GCC-Toolchain** zum Programmieren des Mikrocontrollers.
  - Die Eclipse-Foundation für die Programmierumgebung **Eclipse**, Kees Bakker und Thomas Holland für die Erweiterung **AVREclipse** sowie Ben Gardner für **Uncrustify** und die damit verbundene Übersichtlichkeit des Quellcodes.
  - Peter Dannegger für den robusten, aber dennoch schlanken Bootloader **FastBoot**.
  - Leo-Andres Hofmann für die schöne Windows-Oberfläche **UpdateLoader**.
  - **Subversion** und **TortoiseSVN** bzw. **Git** und **GitHub** für die unkomplizierte Versionsverwaltung.
  - **LATeX** und **TeXstudio** zur Erstellung des Handbuchs.