

El Fueradoro

Funkgesteuertes Zündsystem für Feuerwerke

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. Nie eine Zündbox als Transmitterbox konfigurieren, um unkontrollierte Zündungen und Schäden am Controller auszuschließen.**
- 2. Keine Softwareupdates an Zündboxen durchführen, solange Zünder angeschlossen sind, da das Verhalten nach einem fehlgeschlagenen Update unvorhersehbar ist.**
- 3. Devices nie ohne angeschlossene Antenne einschalten, um Rückreflexionen an offenen Steckern/Buchsen zu vermeiden, welche die Sendeendstufe zerstören könnten.**

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

Inhaltsverzeichnis

1	System	4
1.1	Transmitterbox und Zündboxen	4
1.1.1	Aufbau	4
1.1.2	Grundfunktionen	5
1.1.3	Energieversorgung	7
1.1.4	Die Status-LEDs	7
1.1.5	Das LC-Display der Transmitterbox	7
1.1.6	Die Schalter der Zündboxen	8
2	Vorbereitung des PCs	10
2.1	Installation eines USB-RS232-Adapters	10
2.2	Einrichtung des Terminalprogramms	12
2.3	Einrichtung von <i>Pyro Ignition Control</i>	13
3	Kommunikation zwischen PC und Devices	14
3.1	Das Konfigurationsprogramm	14
3.2	Die Systemübersicht	18
3.3	Manuelles Senden	18
4	Aufspielen des Bootloaders	20
5	Firmwareupdate	22
A	Installation „Prolific PL2303“	24
B	Antennenbau	26
C	Credits	29

1 System

El Fueradoro ist eine Eigenentwicklung zur automatisierten Zündung von Feuerwerkschoreographien per Computer und Funk, der Name *El Fueradoro* leitet sich 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) geplant.

Es besteht aus einer mit dem PC verbundene Transmitterbox, welche die von *Pyro Ignition Control* generierten Zündbefehle per Funk an die Zündboxen weitergibt und einer oder mehreren Zündboxen mit je 16 Kanälen zur elektrischen Zündung. 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.

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 (Slave-ID und Unique-ID) zugewiesen werden. Genaueres dazu findet sich im Abschnitt 3.

Transmitterbox und Zündbox sind in Abbildung 1.2 bzw. 1.3 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.

1.1 Transmitterbox und Zündboxen

1.1.1 Aufbau

Herzstück aller Devices ist ein 8-bit-Mikrocontroller von Atmel, unterstützt werden der ATmega168PA sowie sein großer Bruder ATmega328P, welcher die verschiedenen Peripheriebausteine kontrolliert. Die Belegung für die verschiedenen Devices ist in Abbildung 1.1 gezeigt. In jedem Device befinden sich ein über eine SPI-Schnittstelle steuerbares Funkmodul von HopeRF, unterstützt werden das inzwischen abgekündigte aber immer noch weit verbreitete RFM12(B) sowie dessen Nachfolger RFM69CW, welches die drahtlose Kommunikation zwischen den einzelnen Devices übernimmt, ein RS232-Treiberbaustein mit an die Gehäuseaußenseite geführter Sub-D-Buchse zur Kommunikation zwischen Mikrocontroller und PC sowie vier Status-LEDs.

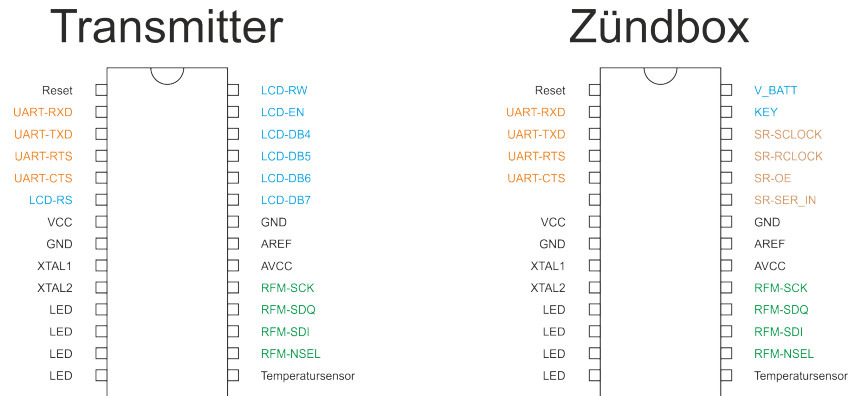


Abbildung 1.1: Pinbelegung des Mikrocontrollers bei Transmitter und Zündbox

Die Transmitterbox ist mit einem LC-Display ausgestattet.

In den Zündboxen befinden sich zwei vom Mikrocontroller gesteuerte kaskadierte Schieberegister, welche mit ihren insgesamt 16 Ausgängen Feldeffekttransistoren für die Zündung ansteuern, außerdem ist auf der Platine ein Hochsetzsteller zur Erzeugung einer höheren Zündspannung integriert sowie ein Schlüsselschalter zum Scharfschalten der Boxen. Parallel zur Drain-Source-Strecke jedes Feldeffekttransistors ist eine LED mit Vorwiderstand geschaltet, um zu signalisieren, an welchen Kanälen Zünder angeschlossen sind.

1.1.2 Grundfunktionen

Die in Abbildung 1.2 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 Bundesnetzagentur – frei nutzbare Frequenzband um 868 MHz verbaut. Die Datenübertragung vom PC geschieht über eine serielle Schnittstelle.

Die Zündboxen, in Abbildung 1.3 dargestellt, empfangen die Anweisungen und zünden die einzelnen Kanäle. Jede Zündbox verfügt über 16 einzeln ansteuerbare Zündkanäle. Jeweils eine rote und eine schwarze Klemme bilden 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. Parallel zur üblicherweise sperrenden Drain-Source-Strecke jedes MOSFETs ist eine grüne LED mit Vorwiderstand geschaltet, welcher den Strom über 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 LED signalisiert.

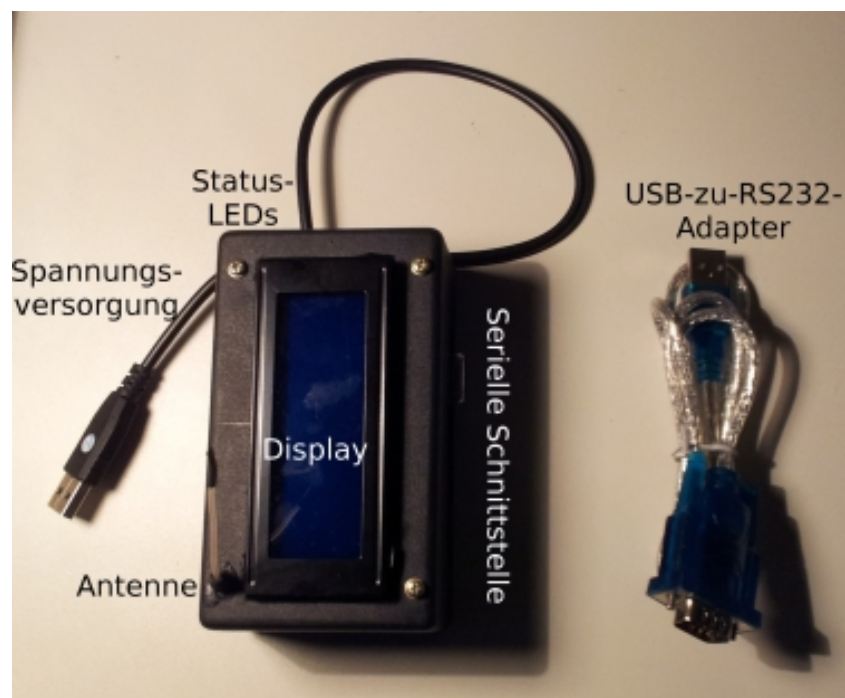


Abbildung 1.2: Transmitter mit zusätzlichem RS232-Adapter



Abbildung 1.3: Zündbox von oben, eingebettet in Koffer

Farbe	Funktion
blau/orange	Funkmodul empfängt
grün	Funkmodul sendet
gelb	Daten kommen über serielle Schnittstelle an
rot	Zündbox ist scharf geschaltet

Tabelle 1.1: Farben und Funktionen der Status-LEDs

1.1.3 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 1.3 über der Zündbox zu erahnen ist.

1.1.4 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. Das Blinken der grünen LED während des Initialisierungsvorgangs jedes Devices ist von der sonstigen Funktion unabhängig.

Bei einem Zündvorgang leuchten für die Dauer der Zündung (11 ms) alle vier LEDs.

1.1.5 Das LC-Display der Transmitterbox

Aktuelle Statusanzeigen werden bei der Transmitterbox auf dem LC-Display ausgegeben. Ein Beispiel zeigt Abbildung 1.4. In Zeile 1 wird hinter der Abkürzung „Tx“ (Transmitted) der letzte gesendete Befehl (Zündbefehl oder Identifizierungsaufforderung) angezeigt, in Zeile 2 hinter



Abbildung 1.4: LCD während der Show

der Abkürzung „Rx“ (Received) die letzte empfangene Rückmeldung (angeforderte Parameter, Wiederholungsaufforderung, Fehlermeldung).

In den Zeilen 3 und 4 werden die letzten sechs gesendeten Kommandos aufgelistet. Zwei zweistellige Zahlen getrennt durch einen Doppelpunkt 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“. 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.

1.1.6 Die Schalter der Zündboxen

Die Zündboxen verfügen, wie in Abbildung 1.5 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.

Ob Boxen scharf geschaltet sind, ist an der Box – wie in Abschnitt 1.1.4 ausgeführt – durch das Dauerleuchten der roten Status-LED erkennbar, kann aber auch durch eine Identifizierungs-



Abbildung 1.5: Schalter und serielle Schnittstelle an der Zündbox

abfrage ausgelesen werden (siehe Abschnitt 3.1).

2 Vorbereitung des PCs

Zur Kommunikation mit einem Computer verfügen alle Devices über eine serielle Schnittstelle. *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 1.2 mit USB-Anschluss für den Rechner und einem 9-poligen RS232-Stecker für den Anschluss an der Peripherie, also die Devices von *El Fueradoro*. In diesen Adapterkabeln ist ein Chip verbaut, um die Signalumsetzung von USB auf RS232 und umgekehrt zu bewerkstelligen. Übliche verwendete Chips sind der CH340¹, welcher sich in vielen über eBay aus China angebotenen Modellen befindet, der Prolific PL2303² 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. Für die Prolific-Treiber ist dies im Anhang A dokumentiert. Hat man die Installation erfolgreich absolviert, sollte bei angeschlossenem Adapterkabel ein neuer Eintrag in der Art von Abbildung 2.1 im Gerätemanager auftauchen.

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

²Falls 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

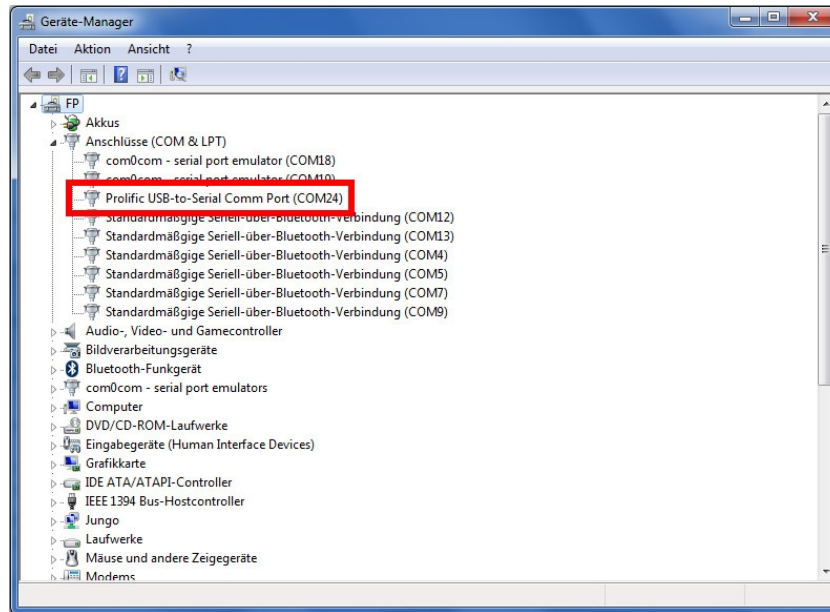


Abbildung 2.1: Eintrag des USB-RS232-Adapters im Gerätemanager

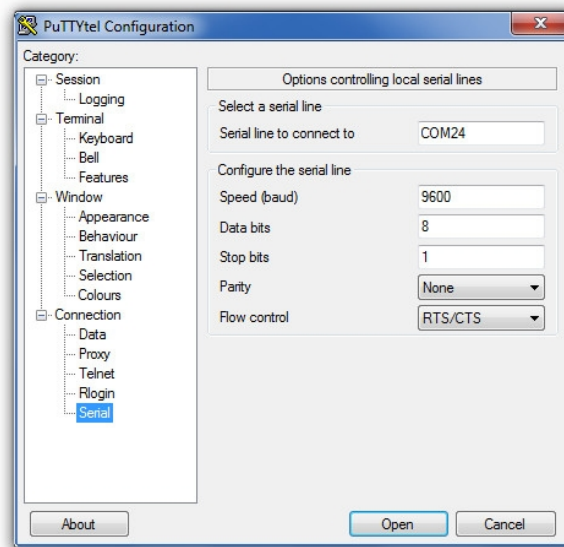
Weil viele Chips dazu tendieren, sich beim Anschluss an immer wieder andere USB-Ports neu zu installieren bzw. eine andere COM-Port-Nummer anzunehmen, wird empfohlen, für den USB-RS232-Adapter stets denselben USB-Steckplatz zu nutzen.

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. Um das Feature Firmwareupdate, welches in Kapitel 5 genau beschrieben wird, nutzen zu können, sollte die Portnummer 1, 2, 3 oder 4 lauten.

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

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*³, 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 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, ehe man die serielle Verbindung per Klick auf „Open“ startet. Vor dem Start der Verbindung sollte man das verbundene Device mit Strom versorgen.

Um bei *Puttytel* nicht immer all diese 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.

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

2.3 Einrichtung von *Pyro Ignition Control*

Auf die Verwendung von *Pyro Ignition Control* soll an dieser Stelle nicht allzu ausführlich eingegangen werden. Wichtig ist allerdings, dass unter „Einstellungen → Optionen“ im Reiter „Output“ der richtige COM-Port eingestellt ist. Außerdem sollte im Reiter „Allgemein“ die globale Verzögerung erfahrungsgemäß auf etwa 0,05 s eingestellt werden, da dies die Zeit ist, die in der Zündbox zwischen dem Empfangen des Zündbefehls und dem Auslösen vergehen. Als minimale Zeitdauer zwischen zwei Zündungen sollten 100 ms nicht unterschritten werden. Über „Einstellungen → Connect“ wird die serielle Verbindung aufgebaut.

Anmerkung: Eine serielle Verbindung kann immer nur **ENTWEDER** zwischen *Puttytel* und dem Device ODER zwischen *Pyro Ignition Control* und dem Device bestehen. Man muss also immer die bestehende Verbindung trennen, bevor man mit dem anderen Programm eine neue aufbauen kann.

3 Kommunikation zwischen PC und Devices

In diesem Abschnitt geht es um die Systemüberwachung bzw. -konfiguration mittels *Puttytel*. Auf die Kommunikation zwischen *Pyro Ignition Control* und der Transmitterbox wird an dieser Stelle nicht eingegangen, da hier alles quasi-automatisch und ohne Zutun des Benutzers stattfindet.

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, dürfen zwischen der Eingabe der einzelnen Buchstaben nicht mehr als 1,7s 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 1,7s später dennoch ausgelöst. Möchte 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 funktionieren lediglich zwei nicht bei allen Devices. „orders“ setzt voraus, dass das angeschlossene Device zuvor – über das durch „conf“ zu erreichende Menü – als Transmitterbox konfiguriert wurde bzw. im Fall von „zero“ nicht als Transmitterbox.

3.1 Das Konfigurationsprogramm

Mit „conf“ gelangt man ins Konfigurationsmenü, welches den Benutzer wie in Abbildung 3.1 gezeigt empfängt. Hier sind die beiden wichtigsten Parameter jedes Devices, die Slave-ID und die Unique-ID, aufgeführt, die der Benutzer nach eigenen Bedürfnissen vergeben kann.

Die **Unique-ID** dient der Identifikation jeder einzelnen Zündbox im Funksystem. Jeder ver-

Befehl	Wirkung
conf	Startet das Konfigurationsprogramm zur Zuweisung von Slave- und Unique-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 gleichzeitig alle anderen Devices auf, ebenfalls eine Temperaturmessung durchzuführen. Zum Auslesen der neu gemessenen Temperaturen muss noch eine Identifizierungsanfrage geschickt werden
list	Zeigt die Systemübersicht (Zuweisung Unique- und Slave-ID, Batteriespannung jeder Box, Scharfschaltungsstatus Temperatur, RSSI, Anzahl Boxen je Slave-ID)
orders	Gibt letztes gesendetes und empfangenes Pattern auf LCD aus
zero	Markiert alle Kanäle der verbundenen Zündbox als noch nicht abgefeuert
cls	Löscht den Terminal-Bildschirm
kill	Löst einen Neustart des Device aus

Tabelle 3.1: Kommandos zur Konfiguration über die serielle Schnittstelle

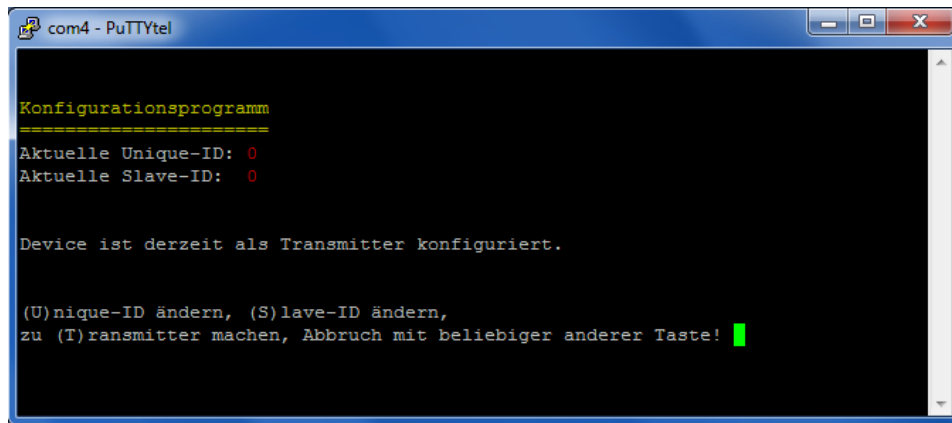


Abbildung 3.1: Startbildschirm des Konfigurationsprogramms

wendeten Zündbox muss daher, um die Funktion von *El Fueradoro* gewährleisten zu können, eine andere Unique-ID im Bereich von 01-30 (zweistellige Eingabe!) zugeteilt werden.

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. Wie oben ausgeführt muss die Zuweisung der Unique-IDs allerdings eineindeutig sein, d.h. jede Box hat eine unterschiedliche Unique-ID bzw. jede Unique-ID gehört zu genau einer Zündbox.

Die Null als Slave- und Unique-ID identifiziert ein Device als Transmitterbox.

Die Zuweisung von Slave- und Unique-ID vom Startbildschirm des Konfigurationsprogramms aus geschieht, indem man den Anweisungen auf dem Bildschirm folgt. U bzw. u (Groß- oder Kleinschreibung spielt hier keine Rolle) ändert die Unique-ID, S bzw. s die Slave-ID einer Zündbox. T bzw. t macht das Device zum Sender, indem Unique- und Slave-ID auf 0 gesetzt werden. Die neue Unique- bzw. Slave-ID muss stets zweistellig ohne Bestätigung durch Enter oder eine andere Taste eingegeben werden, die Konfiguration als Transmitter kann nicht über die Eingabe von Unique- und Slave-ID, sondern nur über die Eingabe T bzw. t im Hauptbildschirm erfolgen.

Möchte man also die Unique-ID 5 zuweisen, muss man zunächst „u“ und anschließend „05“ eingeben, für Slave-ID 12 dementsprechend „s“ und „12“. Ein Sonderfall tritt auf, wenn man ein zuvor als Transmitter konfiguriertes Device nun als Zündbox konfigurieren will. Es erscheint dann nach Eingabe der neuen Slave-ID (Unique-ID) automatisch die Eingabeaufforderung für die neue Unique-ID (Slave-ID), da entweder beide IDs den Wert 0 haben müssen (Transmitter) oder beide IDs einen anderen Wert als 0 (Zündbox). Die zugewiesenen IDs werden an drei Stellen im internen Speicher mit Prüfsummen hinterlegt und bleiben auch nach einem Firmwareupdate erhalten.

Wie bereits angesprochen ist es möglich, auch eine Zündbox als Transmitter zu verwenden. Damit diese aber gleichzeitig noch als Zündbox fungieren kann, muss die Slave-ID größer sein

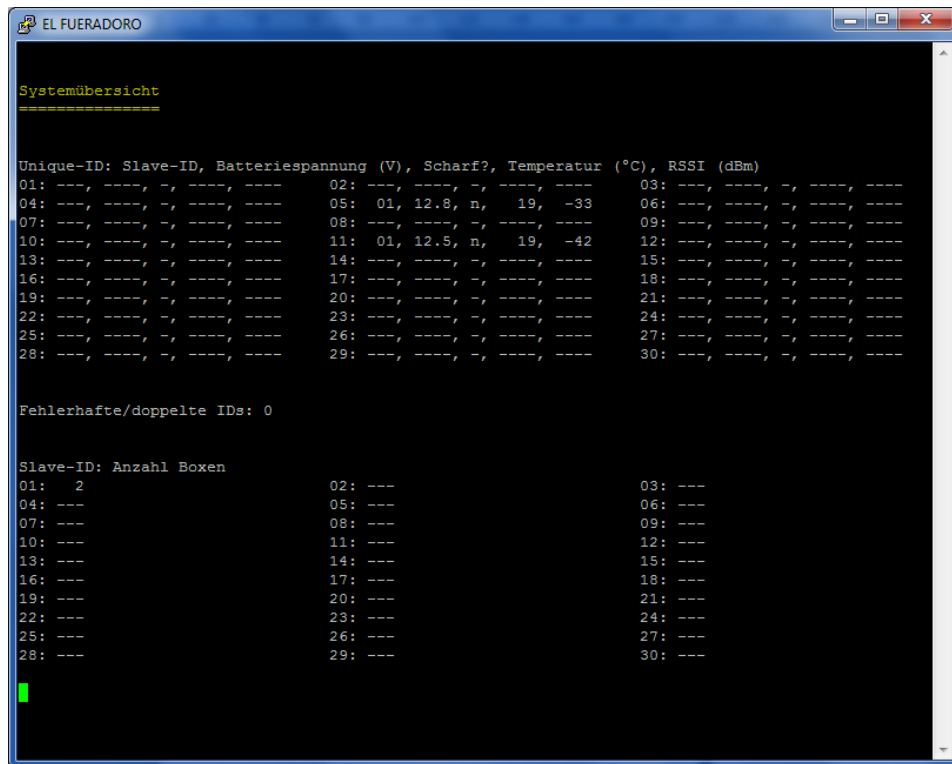


Abbildung 3.2: Systemübersicht

als 0, da mit *Pyro Ignition Control* nur Slaves von 1-30 angesprochen werden können.

WARNUNG: Als Transmitter (Slave-ID 0 und Unique-ID 0) konfiguriert werden dürfen nur Devices wie in Abbildung 1.2 mit Display und ohne Klemmen für den Anschluss von Zündern. Der Grund hierfür liegt in der in Abbildung 1.1 gezeigten Pinbelegung des Controllers. Wie zu erkennen ist, 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. Da die Software davon ausgeht, dass ein Device mit Slave-ID und Unique-ID 0 an diesen Pins ein LC-Display besitzt, werden die Pins entsprechend konfiguriert und angesteuert. Im Fall des Schlüsselschalter-Pins kommt es daher zu einem Kurzschluss, wenn die Box scharf geschaltet ist, was den Controller beschädigen kann, die Ansteuerung des Schieberegisters mit LCD-Befehlen führt dazu dass Zündkanäle durchschalten und Zündungen unkontrolliert ausgelöst werden, weil die Software glaubt, die Datenpins des LCD zu modulieren und nicht die Schieberegister anzusteuern.

Der umgekehrte Fall, eine Transmitterbox als Zündbox zu konfigurieren, ist nicht sinnvoll, wird aber zu keinerlei Schäden führen.

3.2 Die Systemübersicht

Mit „list“ ist es möglich, sich die Systemübersicht entsprechend Abbildung 3.2 anzeigen zu lassen. Es werden zwei Tabellen ausgegeben, wobei die obere nach Unique-ID geordnet anzeigt:

1. Welche Slave-ID der Unique-ID zugewiesen ist.
2. Welche Spannung die Batterie der Box mit der entsprechenden Unique-ID liefert.
3. Ist die Box mit der jeweiligen Unique-ID scharf geschaltet: (j)a oder (n)ein.
4. Temperatur im Inneren der Box, sofern die Box über einen eingebauten Temperatursensor verfügt.
5. Wie stark ist das von der Box empfangene Signal (RSSI = Received Signal Strength Indication). Werte stehen nur bei Aufruf von „list“ an einem Device mit RFM69 zur Verfügung.

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, für eine möglichst aktuelle Liste sollte also zuvor, wie im Abschnitt 3.3 beschrieben eine Identifikationsaufforderung gesendet werden.

3.3 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 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.

Das Senden einer Identifikationsaufforderung hat zudem den Effekt, dass alle Kanäle in allen Zündboxen als noch nicht abgefeuert gekennzeichnet werden.

Jede andere Angabe als „f“, „i“ oder „t“ beendet den Modus ohne irgendetwas zu senden.

Denselben Effekt hat die Eingabe einer Slave-ID oder Kanalnummer außerhalb der jeweils zulässigen Zahlenbereiche.

4 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 Fueradorno*-Firmware normal ausgeführt werden soll.

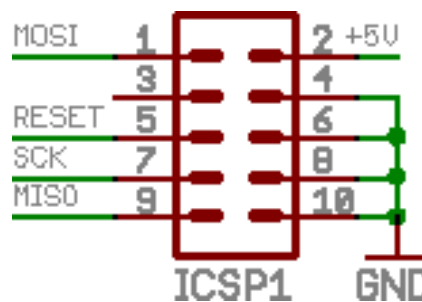


Abbildung 4.1: Pinbelegung des ISP-Platinensteckers (Ansicht von oben, Aussparung an Pin 5)

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, welche den Controller in den Resetzustand versetzt und anschließend den Bootloader über die SPI-Schnittstelle an eine festgelegte Stelle im Flash-Speicher schreibt.

Auf der Platine jedes Devices ist für ISP ein zehnpoliger zweireihiger Wannenstecker angebracht, an den gängige Programmiergeräte wie der weit verbreitete *AVRISP mkII* angeschlossen werden. Seine Pinbelegung ist in Abbildung 4.1 gezeigt.

Zum Brennen des Bootloaders gibt es ein Kommandozeilentool namens *btldflsh.exe* für den *AVRISP mkII*, welches auf *AVRDUDE* basiert. 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:

```
btldflsh.exe bootloader_m328p_9830400.hex
```

Die für *El Fueradorno* benötigten Einstellungen für Fuses und die Datenübertragung werden

auf diese Weise automatisch angepasst. Nach einmaligem Flashen des Bootloaders wird die ISP-Schnittstelle nicht wieder benötigt.

5 Firmwareupdate

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

Die aktuelle Firmware kann aus dem SVN-Repository

<https://code.google.com/p/el-fueradoro/>

heruntergeladen werden. Sie beinhaltet den kompletten C-Quellcode, die kompilierten iHex-Dateien für Firmware und Bootloader, die nötigen Software-Tools zur Übertragung zwischen PC und Mikrocontroller sowie die vorliegende Anleitung.

El Fueradoro bietet die Möglichkeit, die Firmware via serielle Schnittstelle vom PC aus zu aktualisieren. Hierfür gibt es das Programm *fwupdate.exe*, welches zunächst einen Reset auslöst, um den Bootloader des Devices zu aktivieren und anschließend die im iHex-Format vorliegende Firmware überträgt.

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. Aufgrund der Möglichkeit, zwei verschiedene Mikrocontroller und zwei verschiedene Funkmodultypen zu verwenden, existieren vier verschiedene Firmwaredateien, welche sich im gleichen Ordner wie *fwupdate.exe* befinden sollten.

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_RFM69.hex
```

Die Angabe der Dateiendung *.hex* kann hierbei – ebenso wie das *.exe* hinter *fwupdate* – auch weggelassen werden, die Firmware-Datei muss jedoch zwingen 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:

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.

A Installation „Prolific PL2303“

Der PL2303 von Prolific gehört zu den am häufigsten anzutreffenden Chips in USB-RS232-Adaptern. Die Installation gestaltet sich jedoch nicht immer einfach¹, denn es existiert offenbar eine Vielzahl von Varianten, die von Windows durch Plug-and-Play nicht korrekt erkannt werden.

Oft ist es daher so, dass beim Verbinden des Adapters mit dem Rechner dieser zwar den Anschluss eines neuen Geräts registriert und in Windows-Update einen Treiber dafür findet, nach Abschluss der Installation der Chip jedoch im Gerätemanager mit dem ungeliebten Ausrufezeichen auf gelbem Grund aufgeführt ist und nicht funktioniert. Als Fehlermeldung erhält man „Fehler 10: Kann nicht gestartet werden“. Im schlimmsten Fall wird gar kein Treiber gefunden und der Rechner meldet im Gerätemanager „Fehler 28: Kein Treiber installiert“.

Der „richtige“ Treiber, mit welchem die meisten günstig über eBay bezogenen PL2303 zu funktionieren scheinen, ist in der Fußnote in Kapitel 2.1 verlinkt. Das heruntergeladene Archiv erhält eine exe-Datei (PL2303_Prolific_GPS_1013_20090319.exe), die man nach dem Entpacken mit Administratorrechten ausführen sollte. Nach Abschluss des Setup-Programms kann man dann im Gerätemanager bei angeschlossenem Adapter wie in Abbildung A.1 links gezeigt zur Tat schreiten und auf „Treibersoftware aktualisieren...“ klicken.

In den beiden nun folgenden Fenstern wählt man „Auf dem Computer nach Treibersoftware suchen“ sowie „Aus einer Liste von Gerätetreibern auf dem Computer auswählen“. Nun er-

¹siehe hierzu auch:

<http://www.adaptare.de/blog/sowas-wieder-yeti-rs232-adapter-mit-echtem-pl-2303-chip-gesichtet/>

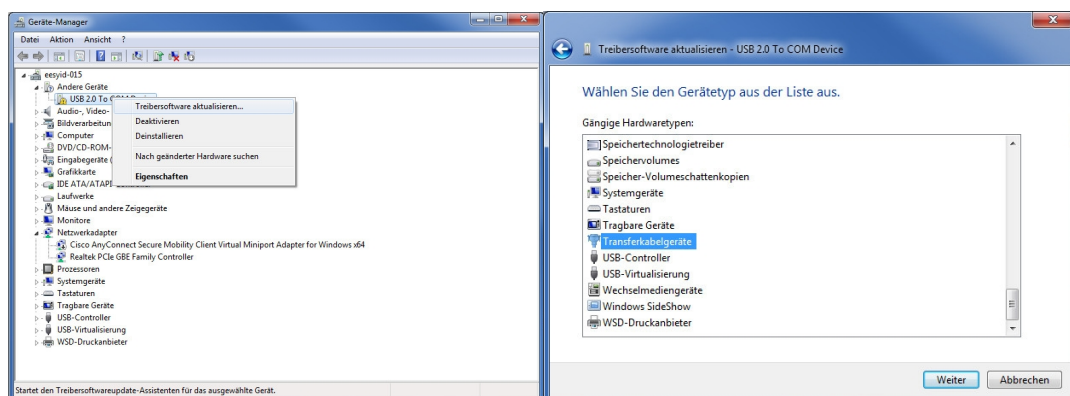


Abbildung A.1: Treiberaktualisierung im Gerätemanager

scheint ein Fenster mit den verschiedenen Gerätetypen, unter denen sich – wie in Abbildung A.1 rechts zu sehen – der Punkt „Transferkabelgeräte“ finden sollte.

Hier ist nun der Treiber „Prolific USB-to-Serial Comm Port Version: 3.3.2.105 [27.10.2008]“ zu wählen.

B 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 Messingröhrchen mit 8 mm Durchmesser und 66,5 mm Länge
- Distanzhülse aus Kunststoff mit 5 mm Länge, 7 mm Außendurchmesser und 3,6 mm Innendurchmesser
- 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 B.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 B.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 verdrehten 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

- 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 10 mm des Mantels entfernt werden, anschließend 5 mm des Schirmgeflechts und zuletzt auch 2,5 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, vercrimpt und Schrumpfschlauch darüber angebracht. Wenn gewünscht kann – wie in B.1 rechts zu sehen – Schrumpfschlauch 9,5 als Witterungsschutz über dem gesamten Steckverbinder angebracht werden.

Die in Abbildung B.2 dargestellte Messung am Netzwerkanalysator zeigt eine sehr gute Anpassung dieser Antenne an $50\ \Omega$ im Bereich um 868 MHz.

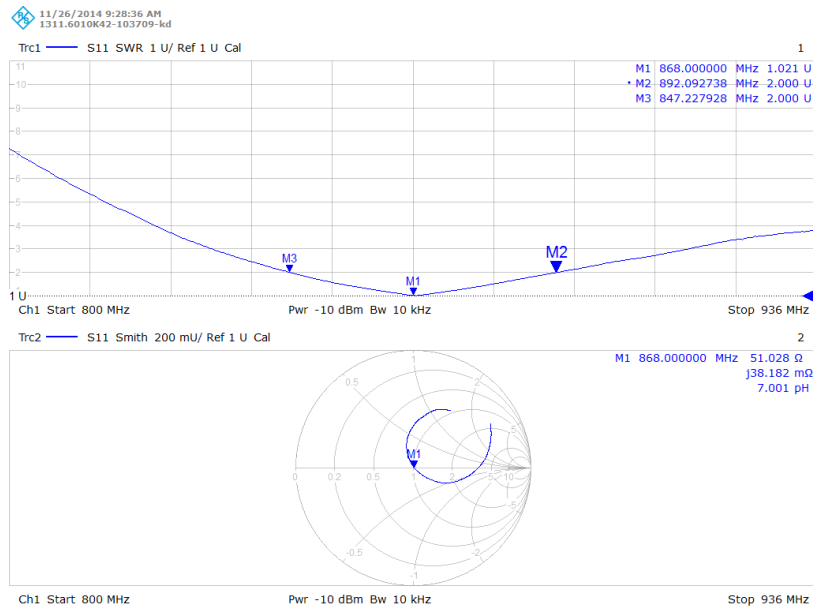


Abbildung B.2: VSWR-Messung der gefertigten Antenne am Netzwerkanalysator

C Credits

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

Als stetige Anlaufstelle für hilfesuchende Elektroniker war die Community von `mikrocontroller.net` eine große Hilfe, ähnliches gilt für `feuerwerk-forum.de`.