

Funkruf-Sender basierend auf Software Defined Radio

Dipl.-Ing. Ralf Wilke DH3WR und Michael Delissen
Amateurfunkgruppe der RWTH Aachen rwth-afu@online.de
am Institut für Hochfrequenztechnik, Melatener Straße 25, 52074 Aachen

Das Funkrufnetz in Deutschland weist an vielen Stellen Versorgungslücken auf. In den letzten Jahren ist der weitere Ausbau nur langsam geschehen. Ein Grund dafür ist die mangelhafte Verfügbarkeit von neuer Sender-Hardware. Gewöhnlich werden umgebaute Ericsson C 9000 als Funkrufsender verwendet. Deren Gewicht, Größe und Stromverbrauch von ca. 50 W schränken ihre Verwendung ein. Der Bezug der C 9000 über Flohmärkte oder im Internet unzuverlässig. Zusätzlich wird eine Controller-Karte, die die Funkrufsignale aus einer Datenkommunikation per RS232 erzeugt, benötigt. Hier gab es zwei Bezugsquellen: ADACOM e.V. mit einem Computer-Design, welches den Möglichkeiten von vor 20 Jahren entspricht und eine Mikrocontroller basierte Variante vom AATiS e.V. Letztere ist aber nicht mehr lieferbar und der Entwickler veröffentlicht den Quellcode nicht. Um diesen Mangel an einfach beschaffbaren Funkruf-Sendern auszugleichen, wurde eine Soundkarten-Lösung programmiert. Eine in Java geschriebene Software kommuniziert über TCP mit den Funkruf-Mastern und erhält so die Daten der Funkrufe. Über die Soundkarte wird das fertig kodierte POCSAG-Signal erzeugt und auf den Modulationseingang eines 9k6-fähigen Senders gegeben. Die Software läuft unter Windows und Linux. Die Konfiguration erfolgt leicht über eine grafische Oberfläche. Mittels eines Suchlaufs kann automatisch der benötigte NF-Pegel gefunden werden.

Geschichte der Funkruf-Sender in Deutschland

Als in den 1990er Jahren das Packet-Radio-Netz noch gut ausgebaut und funktionstüchtig war, verbreitete sich eben dem Punkt-zu-Punkt-Datenfunk der AX.25-Verbindungen ein weitere digitale Betriebsart im Amateurfunk. Die heute selbstverständlichen Mobiltelefone verdrängten gerade ihren Vorgänger, die Funkruf-Empfänger oder mit englischer Bezeichnung 'Pager'. Diese zunächst numerischen, später alphanumerischen tragbaren Empfangsgeräte mit den Markennamen Skyper oder Quix sind batteriebetrieben und haben ein LC-Display. So können sowohl persönliche Nachrichten empfangen als auch sogenannte Rubriken abonniert werden. Das System ist ein Rundfunk-System, bei dem ein dichtes Netz von Gleichwellen-Sendern die Versorgung abdeckt. Die typischen Sendeleistungen der im 70cm-Band betriebenen Basisstationen liegen bei 200 W. Findige Funkamateure sind auf die Idee gekommen, diese nun günstig am Markt erhältlichen Funkruf-Empfänger auf Amateurfunkfrequenzen umzubauen. Die damals ausgemusterten kommerziellen Sender waren meist Ericsson C-9000. Neben der Änderung der Sendefrequenz musste auch eine neue Platinenkarte eingebaut werden, welche die auszusendenden Signale generiert. Diese war über eine RS232-Schnittstelle an den RMNC oder Digipeater-Rechner angeschlossen. Über das KISS-Protokoll geschah die Anbindung an das Packet-Radio-Netz, um die Sender mit Daten zu versorgen. Die Controller-Software bestand zunächst aus einer undurchsichtigen Ansammlung von Skripten, welche später durch DH4DAI und DH6BB in den sogenannten FunkrufMaster [1] überführt wurden. Diese eigenständige Linux-Software wurde an zentralen Digits in Deutschland installiert und tauschte über das Packet-Radio-Netz untereinander Informationen aus. Dazu zählte ein Weiterleiten der persönlichen Funkrufe und die Synchronisation der überregionalen Rubriken. Außerdem wurden die geografisch verteilten Sender von der FunkrufMaster-Software zeitlich synchronisiert und mit Daten versorgt. Da das System nur auf einer Frequenz arbeitet, ein Gleichwellen-Betrieb aber nicht erreichbar ist, wurde TDMA (Time

Division Multiple Access) als Zugriffsverfahren gewählt. Das heißt, innerhalb überlappender Versorgungsbereiche müssen die Sender zeitlich so synchronisiert werden, dass zu jeder Zeit immer nur ein Sender in Betrieb ist. Insbesondere vor der Einführung von NTP (Network Time Protocol) war die Uhrzeit der Digi-Rechner teils so verstellt, dass ein sinnvoller Betrieb des Netzes nur schwer möglich war. DCF77-Dongle sollten damals Abhilfe schaffen. Bei einer Aufteilung einer Minute in 16 Zeitschlitze muss die Rechner-Uhr auf die Sekunde genau eingestellt sein, sonst interferieren die Aussendungen.

In der ersten Dekade des Jahrtausends begann der Verfall des Packet-Radio-Netzes. Zwei der Hauptgründe sind die Konkurrenz des Internet und die Beschränkung bei DFMG-Standorten. Auch das Funkruf-Netz war von diesen Einschnitten mitbetroffen. Mit dem Aufschwung durch Hamnet hoffen wir, dass auch das Funkruf-Netz in Deutschland wieder an Attraktivität gewinnt und weiter ausgebaut wird.

Aktueller Status bei den Empfangsgeräten

Durch die ubiquitär verfügbaren Datendienste im Mobilfunknetz ist die kommerzielle Nutzung der Funkruf-Empfänger fast ganz eingestellt worden. Börsennachrichten empfängt man heute auf dem Smartphone per App und nicht mehr auf dedizierten Geräten. Daher sind insbesondere die Empfänger des Typs Skyper bei Online-Auktionshäusern günstig zu erstehen. Im Wesentlichen beschränkt sich der Umbau auf den Tausch des Schwingquarzes für die Empfangsfrequenz sowie die Überbrückung eines Quarz-Bandpassfilters. Letzteres verschlechtert natürlich die Empfangsselektivität des tragbaren Gerätes; eine Neuankunft eines passenden Filters ist aber im Amateurbereich nicht zu bezahlen. Nach einem Abgleich des Antennenkreises auf die neue Frequenz von ursprünglich 465,970 MHz auf 439,9875 MHz ist der Umbau abgeschlossen. Als Bezugsquelle für die Quarze liefern Internet-Suchmaschinen einige Einträge.

Aktueller Status bei den Sendern

Auf der Infrastruktur-Seite ist die Beschaffungslage weitaus schlechter. Im Wesentlichen werden die Ericsson C-9000-Sender verbaut. Diese sind aber nur mit Glück im Internet oder auf Amateurfunk-Flohmärkten zu erstehen, dann meist mit Preisen über 100 Euro. Weitere Nachteile sind der hohe Stromverbrauch von 50 Watt bei 24 Volt, das hohe Gewicht und die Abmaße von 3 Höheneinheiten in 19 Zoll-Bauweise. Dies macht aus heutiger Sicht die C-9000 unattraktiv.

Der nächste Punkt wiegt noch schwerer. Zum Betrieb der C-9000 wird eine RPC-Karte benötigt. Dieser Radio-Paging-Controller kommuniziert in der ursprünglichen Variante über einen KISS-Link auf RS232 mit einem AX.25-Digi. Auf der anderen Seite wird das Modulationssignal und die Sendertastung erzeugt. Für diese Hardware-Komponente gab es zwei Alternativen:

RPC-Karte von ADACOM e. V.

Der ADACOM e.V. hat eine Platine in Größe einer Europakarte entwickelt. Entsprechend dem technischen Fortschritt zur Entwicklungszeit wird hier im Wesentlichen ein Mikrocontroller in dezidierten Hardware-Bausteinen realisiert. CPU, RAM, EPROM usw. finden auf der Platine ihren Platz. Als Datenschnittstelle dient eine RS-232. Aus heutiger Sicht ist der Nachbau dieser Schaltungstopologie nicht mehr zeitgemäß. In Mikrocontrollern mit Preis unter 10 Euro findet man auf 1 cm² die gleiche Funktionalität wie auf der Europakarte. Zudem ist die Verbreitung von EPROM-Brennern weiter zurückgegangen, so dass ein einfacher Nachbau nicht möglich ist. Zusammengefasst ist das Design einfach zu alt, um ernsthaft noch einmal aufgebaut zu werden.

Funkruf-Controllers uPSDrpc von AATiS e.V. (AS637)

Der Arbeitskreis Amateurfunk und Telekommunikation in der Schule e.V. bewirbt in seinem Rundschreiben Sommer/Herbst 2006 eine Platine von etwa halber Europakartengröße, die die gleiche Funktionalität wie die Lösung von ADACOM enthalten soll. Eine Analyse ergibt, dass hier ein Mikrocontroller benutzt wurde, um die Anzahl der Bauteile stark zu reduzieren. Diese Lösung ist also eine Weiterentwicklung der ADACOM-Lösung, die zum Nachbau geeignet war. Leider treten folgende Probleme auf:

- Der Bausatz ist von AATiS nicht mehr lieferbar.
- Der verwendete Mikrocontroller wird nicht mehr produziert.
- Der Entwickler veröffentlicht laut AATiS den Quellcode seiner Software nicht.

Motivation der Arbeit

Damit gibt es also seit mehreren Jahren keine Möglichkeit, zeitgemäß neue Funkrufsender aufzubauen und Versorgungslücken zu schließen. Hinzu kommt, dass das POCSAG-Protokoll im Internet nur an wenigen Stellen dokumentiert ist. Auch die Kommunikation zwischen FunkrufMaster und Funkruf-Sender ist aus Sicht der Autoren nur durch Analyse des Quellcodes erreichbar. Somit bestand für uns akuter Handlungsbedarf, eine Lösung zur einfachen Installation von neuen Funkrufsendern zu schaffen.

Konzept der vorgestellten Lösung

Der Aufbau des Hamnet bringt eine Erneuerung der verbauten Technologien an Standorten automatisch arbeitender Amateurfunkstationen mit sich. RMNCs werden durch PCs ersetzt und statt KISS und AX.25 werden Ethernet und IP verwendet. Damit ist ein Rechner mit Soundkarte (per USB oder On-board) verfügbar oder leicht zu beschaffen. Auch FM-Sender, die einen ausreichenden Frequenzgang besitzen (9k6-fähig), stellen kein Problem dar. Somit wurde auf Basis des Konzeptes des SDR (Software Defined Radio) ein Programm geschrieben, welches als RPC-Controller wirkt. Die wichtigsten Eigenschaften sind:

- Verwendung von Java als Programmiersprache, d.h. lauffähig unter Linux und Windows
- Grafische Oberfläche zur Konfiguration
- Speichern der so erzeugten Konfiguration in einer Textdatei, so dass auch ein Betrieb von der Kommandozeile aus möglich ist.
- Kommunikation mit dem FunkrufMaster über TCP/IP (zeitgemäß und Hamnet-freundlich)
- Ansteuerung der Sendertastung über serielle Schnittstelle (klassisches Interface)
- Ausgabe der Modulation im POCSAG-Format über die Soundkarte

Um diese Zielvorgaben zu erreichen, musste zunächst das POCSAG-Protokoll und die Kommunikation zwischen FunkrufMaster und Funkruf-Sender (auch als FunkrufSlave bezeichnet) analysiert und verstanden werden. Neben dem Quellcodes des Funkrufmaster war [2] eine gute Quelle. Wir möchten diese Tagung nutzen, um das von uns erarbeitete Wissen der Öffentlichkeit vorzustellen und so für mehr Innovation im Bereich Funkruf im Amateurfunk sorgen.

POCSAG

Grundsätzlich sind Aussendungen eingeteilt in sogenannte Batches, die wiederum eingeteilt sind in Codewords. Zunächst sieht der grundsätzliche Aufbau einer solchen Aussendung wie folgt aus:

Präambel Batch Batch Batch ...

Die hier zu Beginn stehende **Präambel** (*abwechselnde Bits 1 und 0, startend mit 1*; besteht aus 576 Bits) dient dazu, die Skyper mit der Sendefrequenz zu synchronisieren bzw. sie aus dem Standby aufzuwecken. Danach folgt eine nahezu beliebige Anzahl an Batches. Zu beachten ist jedoch, dass die gesamte Sendezeit - beginnend bei der Präambel – maximal 30 Sekunden betragen darf und dann (evtl. nach einer Sendepause) zunächst wieder eine Präambel folgen muss.

Ein **Batch** selbst ist aufgeteilt in acht sogenannte **Frames**, wobei man bei 0 zu zählen beginnt (also existieren Frame 0 - 7). Zu Beginn des Batches steht ein **Synchronisationswort** und danach folgen die Frames. Das Synchronisationswort sieht wie folgt aus (hexadezimal): *7CD215D8*.

Batch								
<i>SYNC</i>	<i>Frame 0</i>	<i>Frame 1</i>	<i>Frame 2</i>	<i>Frame 3</i>	<i>Frame 4</i>	<i>Frame 5</i>	<i>Frame 6</i>	<i>Frame 7</i>

Ein Frame wiederum beinhaltet zwei **Codewords**, die jeweils 32 Bit lang sind (ein Batch besteht somit aus 17 mal 32 = 544 Bits). Hier können entweder Adress- (AW) oder Nachrichtenwörter (MW) stehen. Der Aufbau dieser Wörter ist:

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
<i>AW</i>	0	Adresse																	<i>Fktb</i>	Prüfsumme												
<i>M W</i>	1	Nachricht																			Prüfsumme											

Wie diesem Schema zu entnehmen ist, ähneln sich die beiden Codewords. Zu beachten ist nun folgendes: Ein **Adresswort** beginnt immer mit einer 0 und ein **Nachrichtenwort** mit einer 1. Danach folgen beim Adresswort die oberen 18 (binären) Stellen der Adresse (die 3 LSBs werden anderweitig eingebracht, Erklärung folgt später), gefolgt von 2 **Funktionsbits** (im Schema *Fktb*). Diese geben die Kodierung (numerisch, alphanumerisch, ...) an. Bei einem Nachrichtenwort beinhalten diese 20 Bits die Nachricht selbst. Beide Codewords enden anschließend mit einer Prüfsumme (11 Bits). Deren Berechnung folgt später.

Beim Zusammenstellen eines Nachrichtenwortes gilt es nun noch Weiteres zu beachten. Verwendet wird hier eine 7-Bit-Kodierung, d.h. ein **Zeichen** besteht hier aus 7 Bits, eine **Ziffer** (für die **numerische** Kodierung) aus 4 Bits. Dementsprechend passen in ein Codeword keine 3 ganzen Zeichen und auch nur 5 Ziffern. Nachrichten, die länger sind, werden auf mehrere aufeinanderfolgende Nachrichtenwörter aufgeteilt. Des Weiteren muss ein Batch immer vollständig sein, d.h. wenn ein Batch noch keine 16 Codewords enthält, werden die restlichen Frames mit dem Idle-Codeword (*7A89C197*) aufgefüllt.

Bezüglich der Adresse gibt es nun noch eine Besonderheit. Wie bereits erwähnt, enthält das Adresswort nur die oberen 18 Stellen einer Adresse. Die unteren 3 Stellen befinden sich nicht im Adresswort selbst, sondern ergeben sich durch die **Position des Worts im Batch**. Ergeben die letzten 3 Bits z.B. zusammen eine 3 (im Dezimalsystem), so muss sich das Adresswort im 3. Frame befinden. Hier treten auch keine Probleme auf, da 3 binäre Stellen im Dezimalsystem den Bereich von 0 bis 7 abdecken. Dies entspricht genau der Frameanzahl im Batch.

Kommen wir nun zur **Prüfsumme**. Diese setzt sich aus dem Cycling redundancy check und der Parität zusammen. Zunächst berechnet man entsprechend des CRCs die oberen 10 Bits der

Prüfsumme und anschließend das Paritätsbit. Das Verfahren funktioniert wie folgt:

1. Ausgangspunkt: Codeword aus 32 Bits, wobei die unteren 11 Bits nur aus 0 besteht (hier stehen am Ende des Verfahrens CRC und Parität).
2. Man nimmt das Generatorpolynom, durch das man teilen möchte (entspricht: *ED200000*).
3. Beginnend beim MSB (also höchsten Bit) des Codewords verschiebt man das MSB des Generatorpolynoms soweit nach rechts, dass dieses unter der ersten 1 des Codewords steht.
4. Nun werden Codeword und Generatorpolynom entsprechend der Position mit XOR verknüpft.
5. Solange dieses Vorgehen noch nicht bis einschließlich dem 12. Bit von rechts durchgeführt wurde, wird mit Punkt 3 fortgesetzt das Ergebnis von Punkt 3 als Codeword genommen.
6. Ist Punkt 5 bearbeitet, enthält das Codeword sowohl Adresse/Nachricht als auch den CRC und das LSB sollte noch 0 sein. Nun bestimmt man die Parität.
7. Dafür zählt man alle 1 im Codeword zusammen. Ist die Anzahl der 1 ungerade, wird das LSB auf 1 gesetzt, ansonsten bleibt es 0.

Das Codeword beinhaltet nun sowohl die Daten als auch die vollständige Prüfsumme und ist bereit zum Versenden. Eine Tabelle zur Kodierung der Zeichen gibt es zum Download unter [5].

Kommunikation mit dem FunkrufMaster

Dieser Abschnitt beschreibt den Aufbau der Nachrichten, die zwischen Master und Slave verschickt werden. Diese Nachrichten müssen anschließend vom Slave entsprechend eines anderen Protokolls (Pocsag) kodiert und in Audio umgewandelt werden.

Analyse des Protokolls

Es gibt verschiedene Nachrichtentypen, von denen bisher nur vier als relevant gefunden wurden:

Nachrichttyp 2

Eine Nachricht dieses Typs hat folgenden Aufbau:

Nachricht vom Master	2:IDENT	Bsp: 2:04
Nachricht vom Slave	2:IDENT:TIME	Bsp: 2:04:08

IDENT gibt hier eine Art Identifizierung des Masters an und entspricht der aktuellen Zeit des Masters. Der Slave übernimmt IDENT in seiner Antwort und sendet weiterhin seine aktuelle Zeit mit (TIME).

Die Nachrichten dieses Typs dienen zusammen mit den Nachrichten vom Typ 3 zur Synchronisierung der Zeit zwischen Master und Slave. Dies ist aufgrund der Slots nötig. Mithilfe der Nachrichten dieses Typs ermittelt der Master die Zeitdifferenz zwischen dem Aussenden seiner Nachricht und dem Empfangen der Antwort des Slaves. Dieser Wechsel von Nachrichten erfolgt nach dem Verbindungsaufbau, nachdem der Slave dem Master seinen Namen mitgeteilt hat.

Nachrichttyp 3

Nachrichten vom Typ 3 haben den Aufbau 3:+TIME bzw. 3:-TIME, wobei TIME die Anzahl Zehntelsekunden angibt, um die die Uhr des Slaves verstellt werden soll. + und - entsprechen hier dem Hinzuzählen bzw. Abziehen zu der aktuellen Zeit.

Beispiel-Nachricht: 3:+2B

Nachrichttyp 4

Diese Nachricht gibt an, in welchen Slots der Slave senden darf. Der Aufbau der Nachricht ist 4:SLOTS, wobei SLOTS die hexadezimale Darstellung meint, z.B. 4:02468ACF.

Nachrichttyp 5 und 6

Hierbei handelt es sich um Zeit- (5) bzw. Textnachrichten (6). Hier erfolgt die Bestätigung mit der nächsten Nachrichtennummer. Eine Nachricht des Typs 5 sieht z.B. wie folgt aus (Nachrichten des Typs 6 sehen gleich aus): #03 5:1:9C8:0:094016 130212

Erläuterung: #03 – Hier steht die Nachrichtennummer. Mit der nächsten Nummer wird der Erhalt der Nachricht bestätigt (Antwort-Nachricht: #04 +)

5 – Nachrichtentyp; 5 = Zeit, 6 = Nachricht

1 – Gibt die Sendegeschwindigkeit an (1 steht für 1200 bps)

Adresse 9C8 – Hier steht die Skyper-Adresse des Empfängers (in diesem Fall Broadcast-für die Uhrzeit)

0 – Funktionsbit

094016 130212 – Hier steht die Nachricht bzw. Zeit, in diesem Beispiel Uhrzeit und Datum (getrennt durch 3 Leerzeichen)

Auf den Erhalt einer Nachricht antwortet der Slave zur Bestätigung mit einem +. Dies gilt für alle Nachrichtentypen. Eine besondere Bestätigung erfolgt nur bei Typ 5 und Typ 6.

Verbindungsaufbau zwischen Master und Slave

Beim Herstellen der Verbindung erwartet der Master vom Slave zunächst den Namen, der beispielsweise so aussieht: [uPSDrpc/XOS v1.0-SCP-#2345678]

Daraufhin beginnt der Master Nachrichten des Typs 2 zu schicken und wartet bei jeder Nachricht auf eine Antwort des Slaves. Dieser antwortet mit dem gleichen Nachrichtentyp, jedoch in der Variante des Slaves. Hiermit ermittelt der Master den Zeitunterschied zwischen dem Slave und sich. Sobald der Master genügend Informationen über die Zeitdifferenz gesammelt hat, sendet er eine Nachricht vom Typ 3. Hierin enthalten ist die vom Slave durchzuführende Zeitkorrektur. Der Slave bestätigt diese Nachricht mit einem +.

Sind diese Schritte soweit absolviert, teilt der Master dem Slave nun die zugewiesenen Slots mit. Dies geschieht mit einer Nachricht des Typs 4. Auch hier antwortet der Slave mit einem +.

Die Verbindung ist nun erfolgreich hergestellt.

Beispiel:

Master: 2:13d2

Slave: 2:13d2:0025

Slave: +

Master: 2:13d3

Slave: 2:13d3:0026

Slave: +

Master: 2:13d3

Slave: 2:13d3:0026

Slave: +

Master: 2:13d3

Slave: 2:13d3:0026

Slave: +

[...]

Master: 2:13d8

Slave: 2:13d8:002b

Slave: +

Master: 3:+13ad

Slave: +

Master: 4:01245689ACDE

Slave: +

Beschreibung der Software

Die Software ist unter Linux und Windows lauffähig. Zum Einschalten des Senders über eine serielle Schnittstelle wird die Bibliothek rxtx [3] benötigt. Nach dem einfachen Installieren dieser Zusatzkomponente kann das Programm unter Windows mit Doppelklick und unter Linux mit

```
java -jar FunkrufSlave.jar
```

gestartet werden. Weitere Kommandozeilenparameter für z.B. eine Log-Datei gibt die Hilfe aus. Die Option `-nogui` unterdrückt das grafische Fenster und erlaubt so den Betrieb auch auf Servern, die keine grafische Oberfläche installiert haben.

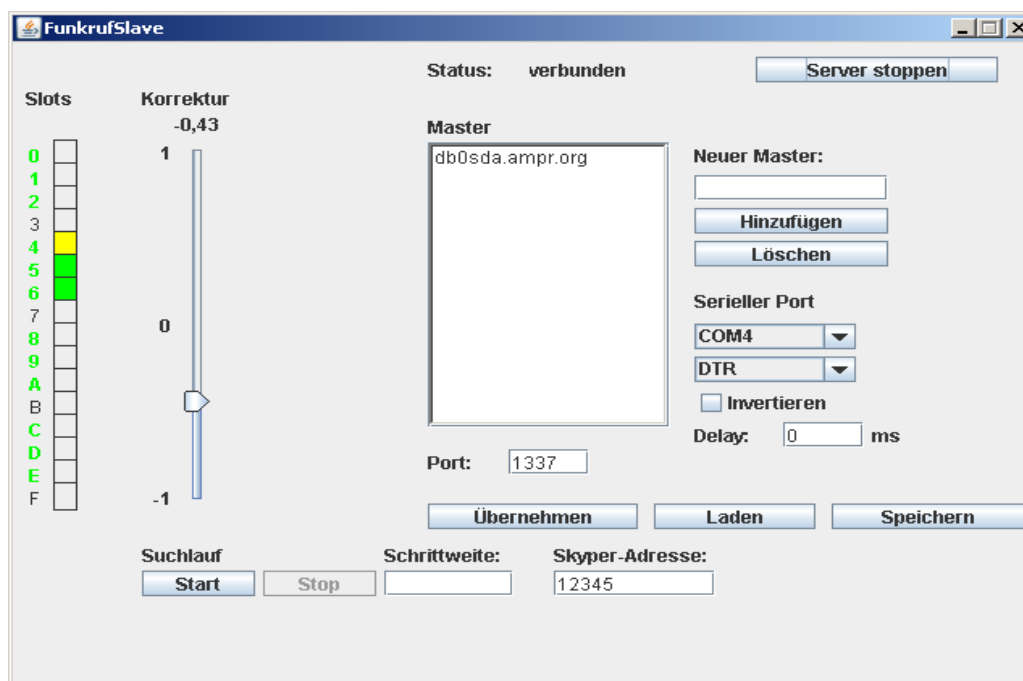


Abbildung 1: Bildschirmfoto der grafischen Konfigurationsoberfläche

Im Fenster können die erlaubten FunkrufMaster eingetragen werden. Dies kann sowohl ein Domainname als auch eine IP-Nummer sein. Meistens wird man nur einen Master eintragen, mehrere sind aber unterstützt. Die Sendertastung wird mittels eines gewöhnlichen RS-232 Interfaces über die Leitungen DTR oder RTS ausgelöst. Sollte die PTT-Beschaltung es verlangen, kann der Schaltpegel auch invertiert werden. Unter Delay ist die Zeit einzustellen, die der Sender vom Einschalten bis zu einem stabilen Betrieb braucht. Um diese Zeit möglichst klein einzustellen wird empfohlen, beim Einschaltvorgang das Sendesignal mittels eines Empfängers und einer Soundkarte aufzuzeichnen. Anschließend schaut man sich mit einem Audioprogramm wie z.B. Audacity [4] den zeitlichen Verlauf des Signals an. Beginnt die Präambel bevor der Sender eingeschungen ist, so muss der Delay-Wert entsprechend erhöht werden. Eine optimale Einstellung dieses Parameters ist für einen hohen Datendurchsatz und ein sicheres Erkennen der Präambel durch die Funkruf-Empfänger notwendig.

Der Frequenzhub bei der verwendeten 2-FSK liegt laut Spezifikation bei 4 kHz. Diese Größe ist direkt vom Signalpegel am Ausgang der Soundkarte abhängig. Wer einen Spektrum-Analysator zur Verfügung hat, kann den Korrektur-Regler genau einstellen. Im anderen Fall kann der automatische Suchlauf verwendet werden. Der Korrektur-Wert ist ein linearer Faktor, mit dem die Lautstärke des erzeugten Ton-Signals und damit auch des Frequenzhub eingestellt werden kann. Negative Werte bedeuten dabei eine Invertierung. Diese Möglichkeit ist notwendig, da nicht alle Funkgeräte bei positiver Modulationsspannung auch eine Frequenzverschiebung in positiver Richtung durchführen. Unter Skyper-Adresse trägt man die Nummer eines vorhandenen Test-Skypers ein, wählt eine

Schrittseite und startet den Suchlauf. Nun werden mit sich änderndem Hub Funkrufe ausgesendet. Inhalt der Nachricht ist der jeweilige Korrektur-Faktor. Wenn der Wert passt, wird der Skyper die Nachricht empfangen. Nun sucht man sich den Mittelwert aller empfangenen Korrektur-Faktoren heraus und stellt den Schieberegler entsprechend ein. Damit ist die Konfiguration abgeschlossen und kann gespeichert werden.

Besteht eine Verbindung mit einem FunkrufMaster, so kann links unter Slots in grün und gelb der zeitliche Verlauf der aktiven und gesperrten Zeitschlitz betrachtet werden.

Die Software kann bei der Amateurfunkgruppe der RWTH Aachen [5] heruntergeladen werden.

Zusammenfassung

Es wurde eine Software vorgestellt, mit der ein Funkruf-Sender einfach aufgebaut werden kann. Neben einer TCP/IP-Verbindung zu einem FunkrufMaster ist nur ein 9k6 fähiges Funkgerät und ein PTT-Interface notwendig. Die Software ist einfach zu konfigurieren und läuft unter Linux und Windows.

Literatur und Quellen

- [1] <http://ham-pager.sourceforge.net/>
- [2] <http://www.uni-stuttgart.de/akafunk/aktivitaet/basics/pocsag.html>
- [3] <http://rxtx.qbang.org/wiki>
- [4] <http://audacity.sourceforge.net/>
- [5] <http://www.afu.rwth-aachen.de/svn/FunkrufSlave/>
- [6] <http://www.uni-kl.de/AFG/funkruf.htm>