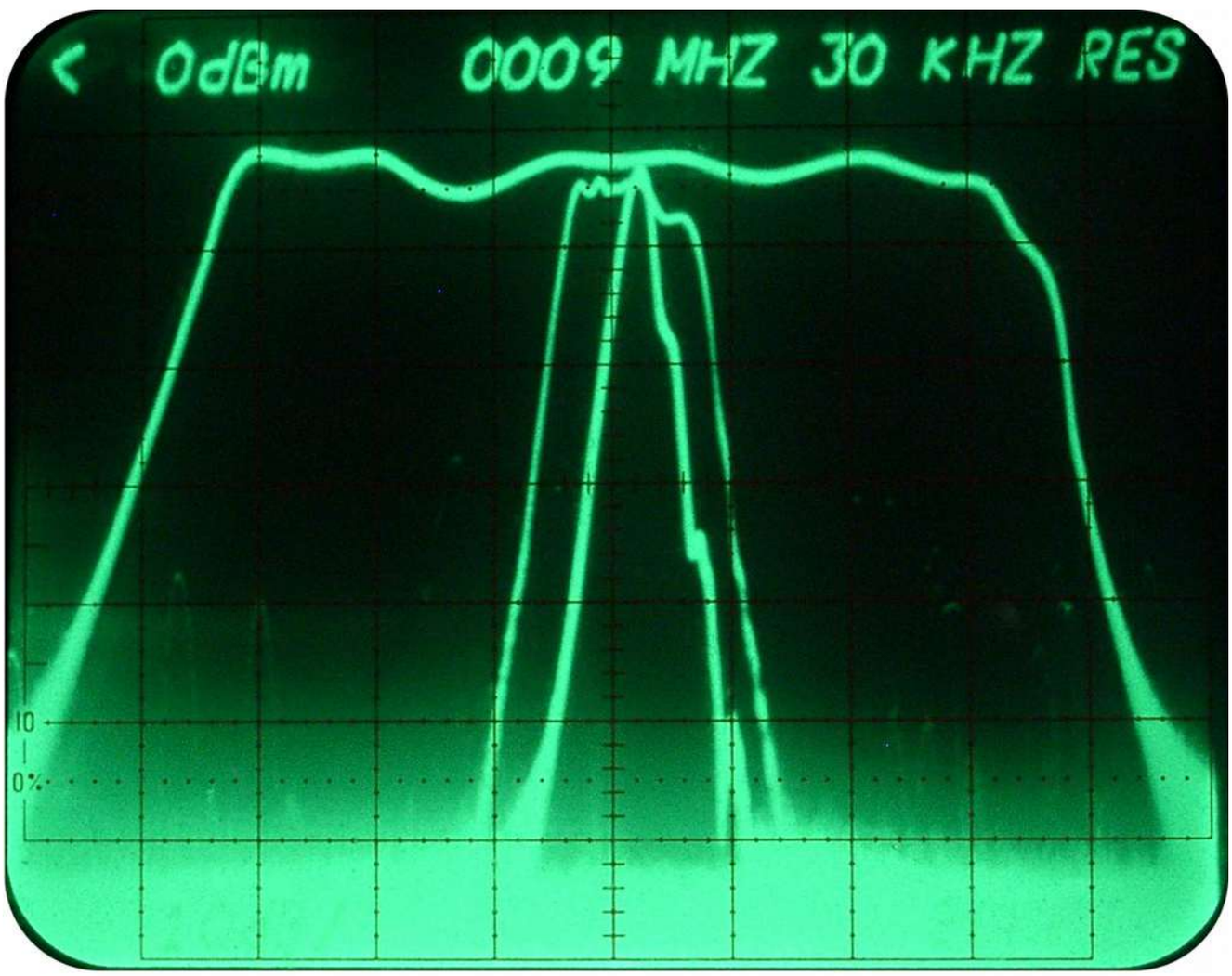
CQ Kronach

Informationen aus dem Ortsverband Kronach
des Deutschen Amateur Radio Club e.V.

Nr. 18 März 2014



CC BY SA

Vorwort

Liebe Mitglieder des Ortsverbandes Kronach,

vielen Dank für das Vertrauen, das mir bei der letzten Hauptversammlung bei der Wahl zum Ortsverbandsvorsitzenden entgegengebracht wurde.

Wir freuen uns, Euch auch in diesem Jahr wieder eine umfangreiche Ausgabe von CQ Kronach, dem alljährlichen Magazin unseres Ortsverbandes, vorstellen zu dürfen. Redakteur Jürgen (DG1NDE) und die Autoren der unterschiedlichen Artikel haben sich mächtig ins Zeug gelegt und eine Reihe allesamt lesenswerter Artikel zusammengestellt.

Wir mussten leider im vergangenen Jahr den schweren Verlust unseres langjährigen Mitglieds Albrecht Täuber (DL6NBL) hinnehmen.

Der Fieldday im abgelaufenen Jahr wurde wieder gut besucht, mein Dank gilt allen, die dazu geholfen haben, dass alles reibungslos ablief.

Auch für 2014 planen wir wieder einen Fieldday, und ich wünsche mir noch die eine oder andere kurzfristige gemeinsame Aktion.

Viel Vergnügen beim Durchlesen, viel Spaß und Erfolg beim DX und in den Wettbewerben wünscht Euch

Frank, DG4NAY
Ortsverbandsvorsitzender

Anmerkung zum Titelbild:

Das Bild zeigt die gewobbelten Durchlasskurven des 9 MHz ZF-Verstärkers des Empfängers, den Manfred (DL9NDK) ab Seite 5 beschreibt. Die äußere Kurve ist die Kurve des FM-Filters mit 12 kHz Bandbreite, die mittlere die des SSB-Filters mit 2,4 kHz Bandbreite und die innere Kurve die des Filters für Morsetelegrafie mit 500 Hz Bandbreite.

CC BY-SA

Inhaltsverzeichnis

Seite	2	Vorwort
	3	Impressum
	4	B21-Vereinsnachrichten
	5	Selbstbauprojekt 2 m-Band Empfänger von DL9NDK
	10	Blick in ~~das Shack~~ den Schuppen bei DG1NDE
	12	Erfahrungen mit der digitalen Betriebsart JT65-HF
	17	Selbstbauprojekt 50 MHz-Transverter von DL9NDK
	22	WSPR-Mobilbetrieb mit Raspberry Pi
	30	Kurzfassung der Creative Commons Lizenz CC BY-SA 3.0
	31	Wir über uns
	31	Termine 2014

CQ Kronach erscheint 1 x jährlich in zur Hauptversammlung des Ortsverbandes Kronach des Deutschen Amateur Radio Club e.V. und wird als Informationsschrift kostenlos abgegeben sowie auf der Homepage des Ortsverbandes http://www.darc.de/distrikte/b/21 im Internet veröffentlicht.

Dieses Werk unterliegt der „Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz“. Um die Lizenz anzusehen, gehen Sie bitte zu http://creativecommons.org/licenses/by-sa/3.0/de/ oder schicken Sie einen Brief an Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA. Auf Seite 30 ist die Kurzfassung dieser Lizenz abgedruckt.

Vereinsanschrift: c/o Frank Sünkel, DG4NAY (1.Vors.)
(zum Zeitpunkt des Erscheinens dieser Ausgabe) Eichenbühl 35
96369 Weißenbrunn
Verantwortlicher Redakteur: Jürgen Fövenyi, DG1NDE

Druck: https://www.wir-machen-druck.de/ Auflage: 100 Stück

WirmachenDruck.de
Sparen Sie bis zu 50% beim Druck!

Da sich durch Neuwahlen Veränderungen der Aufgabenverteilung im Ortsverband ergeben können, wenden Sie sich mit Fragen und Anregungen am besten über das Kontaktformular auf der Ortsverbandshomepage http://www.darc.de/distrikte/b/21/kontakt/ an den jeweiligen Ortsverbandsvorsitzenden. Bezieht sich Ihre Frage oder Anregung auf eine gemäß der Creative Commons Lizenz veränderte oder an anderer Stelle veröffentlichte Version, so wenden Sie sich an den jeweiligen Bearbeiter beziehungsweise Betreiber der Seite, der den Download anbietet.

Obwohl alle Beiträge mit größtmöglicher Sorgfalt zusammengestellt wurden, lassen sich Fehler natürlich nie ausschließen. Für eventuelle Hinweise sind wir dankbar, jegliche Haftung ist jedoch verständlicherweise ausgeschlossen. Beiträge und Abbildungen ohne namentliche Kennzeichnung (Rufzeichen) stammen von der Redaktion. Die Bestimmungen für die Errichtung/Betrieb von Fernmeldeanlagen sowie die einschlägigen Sicherheitsvorschriften und mögliche Urheberrechte sind zu beachten.

(cc) BY-SA

B21-Vereinsnachrichten

Platzierungen bei ausgewählten Funkwettbewerben im Jahr 2013

Aktivitätswettbewerb Franken (Mai 2013)

Klasse B (KW/SSB, 74 Wertungen)

2.	DL2NEE	7360 Punkte	B21
59.	DL1NGS	276 Punkte	B21

Klasse D (70 cm, 49 Wertungen)

30.	DL2NEE	7450 Punkte	B21
39.	DL1NGS	2130 Punkt	B21

Klasse C (2 m, 66 Wertungen)

52.	DL1NGS	3776 Punkte	B21

In der Vereinswertung erreichte Kronach mit 6 Logs und insgesamt 254 Punkten unter den 23 teilnehmenden Ortsverbänden den 8. Platz im Distrikt Franken. Zum Vergleich: Den ersten Platz erreichte der Ortsverband Coburg (B19) mit 47 Logs und 2420 Punkten.

Siehe auch:

vollständige Ergebnisliste 2013 auf der Homepage von Uwe Scherf (DL9NDS)
URL: http://www.dl9nds.de/Deutsch/Akt_Wettbewerb/Download/Auswertung_2013.pdf

Fieldday

Hier einige Impressionen des Fielddays, der – wie es seit einigen Jahren Tradition ist – auch im vergangenen Jahr wieder auf dem Wachtelberg bei Wildenberg stattfand.

Fotos: Alexander Schramm

CC BY-SA

Selbstbauprojekt 2 m-Band Empfänger von DL9NDK

Was lange währt wird endlich gut, kann man dieses Großprojekt nennen. Bereits Anfang der 80er-Jahre konstruierte Friedrich Krug (DJ3RV), mit seinen Studenten der Universität Erlangen ein nahezu optimales Empfängerkonzept. Zuerst war an einen ZF-Verstärker gedacht der auf drei Bandbreiten umschaltbar sein musste. Für jede dieser Bandbreiten sollte ein mehrstufiges Quarzfilter optimale Verstärkungseigenschaften sichern. Dann sollten für die Betriebsarten AM, FM, LSB, USB und CW getrennte Demodulatoren und ein im Frequenzgang optimierter NF-Verstärker folgen. Alleine diese Stufen erforderten sechs Teile in der Zeitschrift „UKW-Berichte“ in einem Zeitraum von anderthalb Jahren. Da die Entwicklung dieser Baugruppen diesen hohen Zeitaufwand erforderte „tauchten“ in der gleichen Zeitschrift parallel dazu geeignete und ebenso aufwändig konstruierte HF-Eingangsteile auf. In meinem Empfänger befindet sich die Schaltung von Joachim Kestler (DK1OF).

Der fertige 2 m-Band-Empfänger

1996 sah ich im Packet-Radio-Netz, dass ein OM alle vollständig bestückten Platinen sehr preisgünstig anbot, da er den Empfänger auf Grund fehlender Messmittel nicht zum Laufen brachte. Ich erwarb alle Platinen und machte mich an die Arbeit. Nach Austausch einiger fehlbestückter und defekter Bauteile konnte der Abgleich beginnen. Es machte viel Spaß die einzelnen Baugruppen abzugleichen, da sie alle in 50 Ohm-Technik aufgebaut sind und einzeln optimiert werden können.

1998 lief dann der Empfänger einwandfrei, mit auf den Tisch liegenden Baugruppen. Diese wanderten dann in eine große Pappschachtel und mussten auf die Ruhephase der Altersteilzeit des Besitzers warten.

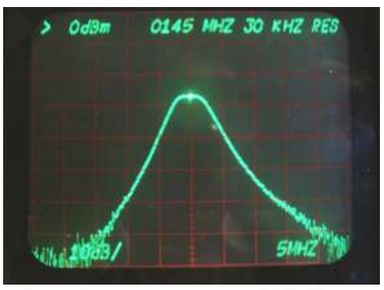
So begann ich 2012 den Bausteinen ein zu Hause zu geben und das Projekt kam 2013 zum Abschluss. Zwar bedient man derartige Geräte heute mit Touchscreen und Microprozessoren übernehmen Steuerungsaufgaben, aber die hochfrequenztechnischen Daten brauchen sich auch heute nicht vor modernen Geräten zu verstecken. Ein herzliches Dankeschön an Detlev (DG6NAI) für die Erstellung der Folie für die Frontplatte.

(cc) BY-SA

Das Konzept:

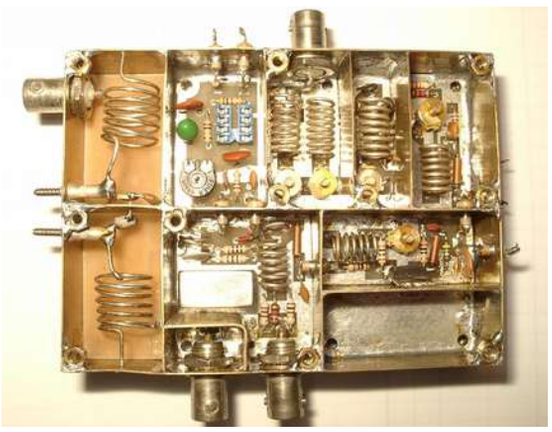
- **HF-Teil:** Helixfilter, GaAs-MESFET, Helixfilter, PIN-Dioden-Dämpfungsglied, Hochstrom-FET, Helixfilter.

HF-Teil mit Helix-Filtern

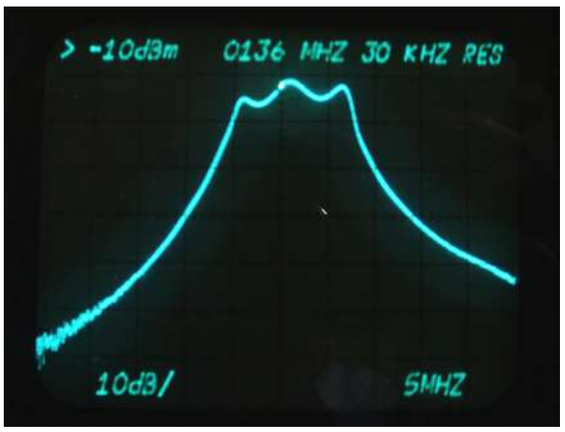


Durchlasskurve des HF-Teils

- **Mischer-Baugruppe:** High-Level-Mischer; das 1 mW-Oszillatorsignal wird eingespeist über ein dreikreisiges Helixfilter, einem MOS-FET und einem Hochstrom-FET, Verstärkung mit HF-Teil: –2 dB bis + 18 dB, Rauschzahl 0,8 dB.

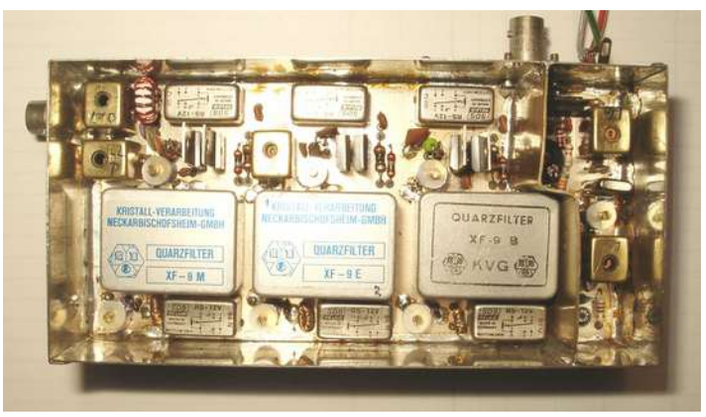


Mischer-Baugruppe mit Helix-Filtern



Durchlasskurve des 136 MHz-Teils im Mischer

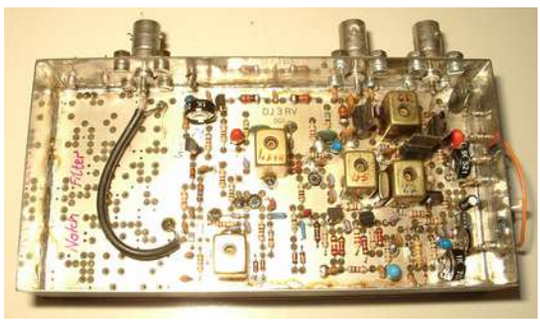
- **9 MHz-Quarzfilter-Baugruppe:** Drei mit Relais umschaltbare achtpolige-Quarzfilter von KVG (12/2,4/0,5 kHz), Anpassverstärker mit Hochstrom-FET. Verstärkung 15 dB, Welligkeit von 2 dB bis 4 dB.



Die 9 MHz-Quarzfilter-Baugruppe (Wobbelkurven siehe Titelbild)

CC BY-SA

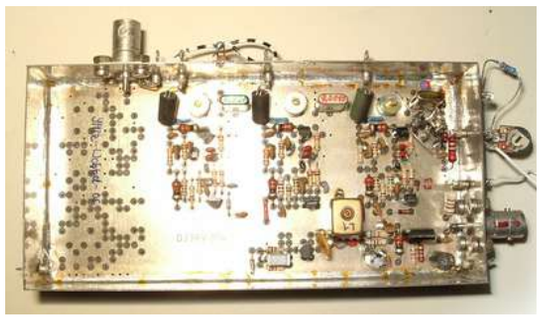
- **ZF-Teil:** MOS-FET, Regelspannungserzeugung, S-Meter-Ansteuerung. Verstärkung 66 dB, Regeldynamik; 100 dB.



9 MHz-ZF-Baugruppe

Demodulator

- **Demodulator-Baugruppe:** Drei getrennte Stufen mit S0 42 P. Squelch-Steuerung, Einspeisung für



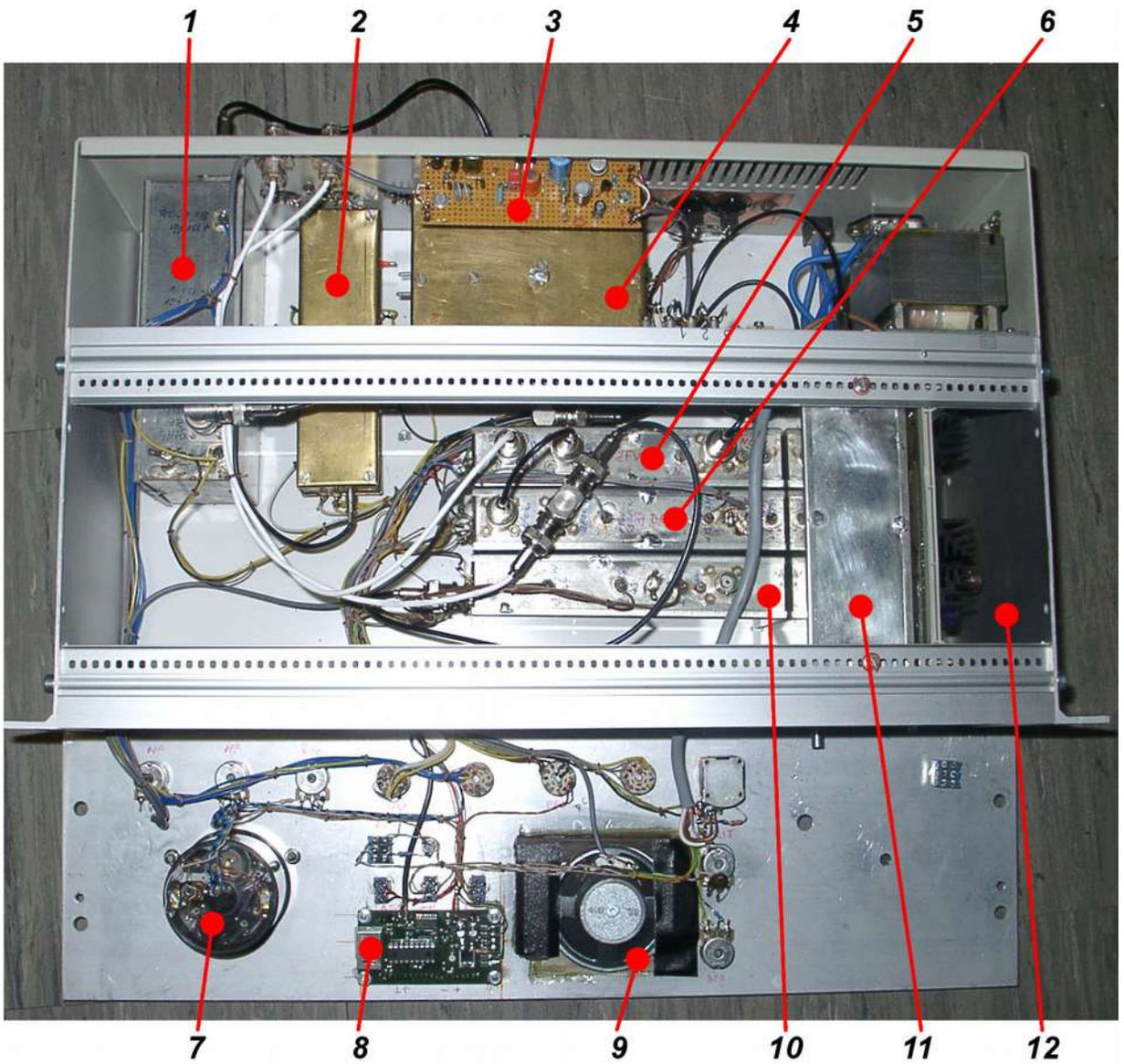
BFO-Baugruppe

- **BFO-Baugruppe:** Drei quarzgesteuerte Trägeroszillatoren für CW, LSB und USB
- **NF-Teil:** Bandpass, TDA 2002, Schaltstufe für Squelch, Ausgangsleistung: 2 W/8 Ohm.
- **VFO:** Hier wurde ein preisgünstig erstandener, früher einmal sehr teurer VFO von Karl Braun (DJ3DT) eingesetzt. Diese geniale Konstruktion arbeitet mit einer Phasenvergleichsschaltung und einer Verzögerungsleitung aus einem PAL-Farbfernsehgerät. Damit entspricht eine 360-Grad-Umdrehung des Abstimmknopfes der Zeilenfrequenz von 15625 Hz. Dies gestattet, ohne Getriebe, eine sehr feinfühlige Abstimmung. Die Frequenzstabilität ist ausgezeichnet.

Der VFO von Karl Braun (DJ3DT)

- **Frequenzanzeige:** Hier wurde ein 2013 neu entwickeltes Modul aus dem Verlag „Funkamateur“ verwendet. Ich begleitete die Entwicklung und trug zur Optimierung des praktischen Einsatzes bei.
- **Netzteil:** Geplant war der Einsatz eines analogen Netzteils. Eine entsprechende Platine war in einem Gehäuse einer ausgesonderten Personenrufanlage, das verwendet werden sollte, vorhanden. Auf dieser Platine wurden drei separate 15 Volt Netzteile aufgebaut, da der Strombedarf, auf Grund vieler eingesetzter Hochstrom-FETs doch recht hoch ist. Leider war die Verlustwärme doch noch so groß, dass die analoge Schaltung einem industriellen Schaltnetzteil weichen musste. Da auch diese Baugruppe relativ warm wurde, kam noch ein kleiner Lüfter zum Einsatz.

(cc) BY-SA



Der geöffnete Empfänger mit abgeklappter Frontplatte:

1: Quarzfilter	*4: Mischer*	*7: Drehspulinstrument*	*10: BFO*
2: HF-Teil	*5: ZF-Teil*	*8: Frequenzanzeige*	*11: VFO*
3: NF-Verstärker	*6: Demodulator*	*9: Lautsprecher*	*12: Netzteil*

CC BY-SA

Praktische Erfahrungen:

Erste Betriebserfahrungen sind sehr zufriedenstellend. Auf der Rückseite des Empfängers sind Einspeisemöglichkeiten für Konverter und ein 9 MHz-ZF-Signal gegeben. So konnte der Empfänger nicht nur auf dem 2 m-Band sondern auch auf dem 6 m- und dem 70 cm-Band mit großem Erfolg getestet werden. Sollten mir noch einige Lebensjahre vergönnt sein, möchte ich ein KW-Teil mit 9 MHz ZF aufbauen und anschließen. Ebenso ist ein 2 m–Sender geplant, der vom RX-VFO gesteuert wird.

Besondere Bedienelemente

Vorderseite: Umschaltung von automatischer auf Handregelung; zusätzlich Umschaltung der Verstärkung des HF-Teiles auf +7, +17, +27 dB. Empfängerfeinverstimmung bei Betrieb mit Sender; 800 Hz Verstimmung von Trägermitte bei CW-Betrieb; grobe Frequenzeinstellung mit „normalen Potenziometer“; Feinabstimmknopf mit 15,625 kHz pro Umdrehung.

Rückseite: Anschlussmöglichkeiten für Zusatzlautsprecher, Kopfhörer, Zusatzgeräte und Antenne; Einspeisung und Auskopplung der 9 MHz-ZF.

Rückseite mit verschiedenen Anschlussmöglichkeiten

Text und Fotos: DL9NDK

CC BY-SA

Blick in ~~das Shack~~ den Schuppen bei DG1NDE

Da sich auch in diesem Jahre wieder bis Redaktionsschluss niemand fand, der im Rahmen der langjährigen Reihe „Blick ins Shack“ seine Funkstation beschreibt, mein eigener Funkraum sich noch immer „in einer Phase der Neuorganisation“ befindet und meine Mobilstation bereits im letzten Jahr ([1]) als Lückenfüller diente, hier in Anlehnung an die ursprüngliche Wortbedeutung des Begriffes „Shack“ einige Impressionen eines Versuchsbetriebs aus einem Geräteschuppen...

> Shack (dt. *Hütte*, *Baracke*), die Bezeichnung für den Ort der Amateurfunkstelle oder sonstigen Funkstation, an dem sich die Sende- und Empfangsgeräte befinden und betrieben werden
>
> *Zitiert aus [2]*

Hier dient eine stillgelegte Wasserleitung als Antennenerde

Der Schuppen und der Garten, in dem mein „Antennentestfeld“ liegt, ist durch eine Hofdurchfahrt (die als einziger Zugang zu einem weiteren Wohnhaus einem Wegerecht unterliegt) von meiner Wohnung getrennt; somit ist eine Kabelverbindung aus dem dort gelegenen „festen“ Shack nicht möglich. Bei den Versuchen zum abgesetzten Betrieb wurde die Datenverbindung in die Wohnung daher über WLAN realisiert.

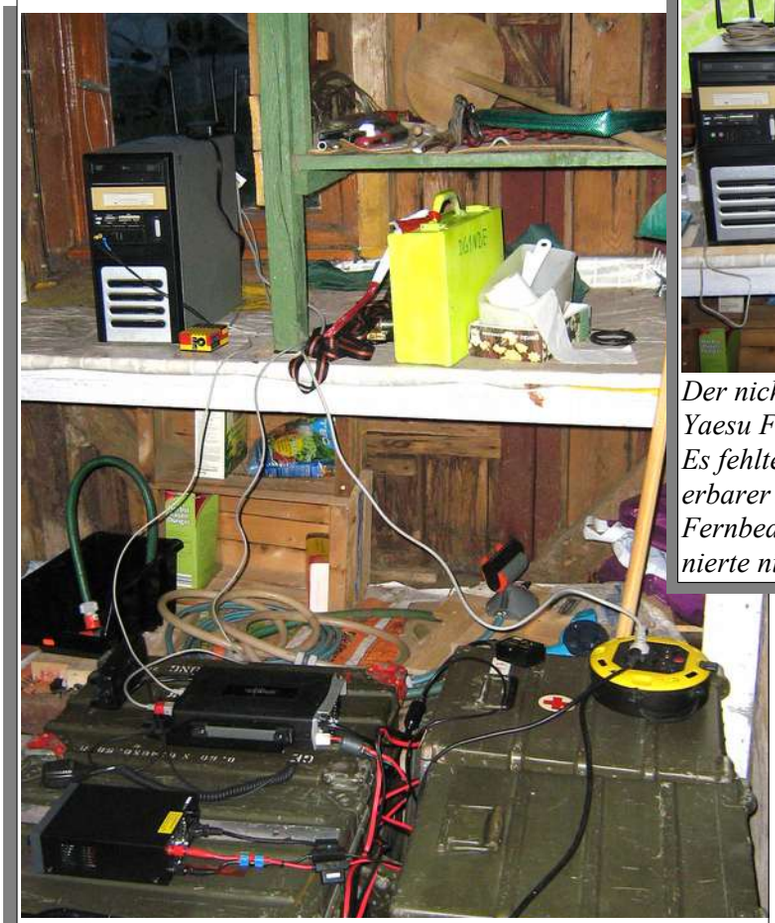
Da die Sendeleistung so angepasst wurde, dass die Hofdurchfahrt außerhalb des standortbezogenen Sicherheitsabstandes beider Stationen liegt, überlappen sich die Sicherheitsabstände von fester Station und Testfeld logischerweise nicht; somit ist in meinen Augen der kurzfristige Aufbau im Schuppen eine eigene, unabhängige Station (auch wenn sie auf dem gleichen Grundstück liegt und daher nur von der Bedieneinheit abgesetzt und nicht fernbedient betrieben wird) und wird von mir bezüglich der Selbsterklärung einzeln betrachtet.

> […] many early stations were built in a garage or tool shed. The term “shack” was only natural and carries through today as a description of the state of order and cleanliness found in many a ham’s lair.
>
> […] viele der frühen Stationen wurden in einer Garage oder einem Geräteschuppen aufgebaut. Die Bezeichnung „Hütte“ war damals also nur folgerichtig, und hat sich bis heute als Beschreibung des typischen Zustandes von Ordnung und Sauberkeit, der in vielen Schlupfwinkeln von Funkamateuren anzutreffen ist, erhalten.
>
> *Zitiert aus [3], sinngemäß übersetzt von DG1NDE*

Steckmasten mit Drahtantenne bei Sonnenuntergang

CC BY-SA

Der nicht allzu erfolgreiche Versuch, einen Yaesu FT-817 fernbedient zu betreiben. Es fehlte nicht nur ein aus der Wohnung steuerbarer Antennentuner, auch die für die Fernbedienung benutzte Software funktionierte nicht immer wie gewünscht.



Der Computer zur Ansteuerung des Funkgerätes, Soundkarteninterface, Transceiver Kenwood TS-480SAT sowie Netzteil. Dieser Versuch war recht erfolgreich, die kostenlos erhältliche Fernbedienungssoftware von Kenwood funktionierte ohne Probleme, nur die Audioübertragung bei verrauschten Signalen war durch digitale Artefakte alles andere als ohrenfreundlich.

Da die abgesetzte Bedienung des FT-817 nicht so gut lief, schloss ich doch noch Monitor, Tastatur und Trackball an. Hier läuft gerade WSPR im Empfangsbetrieb. (Zu WSPR siehe auch den Artikel „WSPR-Mobilbetrieb mit Raspberry Pi“ ab Seite 22.)

Text und Bilder: DG1NDE

Quellen:

[1] Die Mobilstation von Jürgen (DG1NDE); DG1NDE; CQ Kronach Nr. 17/2013; Downloadadresse: http://www.darc.de/uploads/media/cqkc17.pdf

[2] Begriffsklärungs-Seite „Shack“. in: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 30. März 2013, 08:03 UTC. URL: http://de.wikipedia.org/w/index.php?title=Shack&oldid=116200263 (Abgerufen: 27. Februar 2014, 16:37 UTC)

[3] Ham Radio For Dummies, Seite 17; H. Ward Silver; Wiley Publishing, 2004; ISBN 0-7645-5987-7

(cc) BY-SA

Erfahrungen mit der digitalen Betriebsart JT65-HF

Die meisten Funkamateure kennen von den digitalen Betriebsarten nur die Betriebsarten RTTY, PSK31, PSK63, SSTV und vielleicht noch QPSK, Hell, Pactor sowie MFSK.

Besonders für Funkamateure, die nur eingeschränkt arbeiten können, weil sie schlechte Antennenmöglichkeiten haben und auch nur mit kleiner Leistung arbeiten können, gewinnen die digitalen Betriebsarten immer größere Bedeutung.

Auch für die älteren Amateure, bei denen vielleicht die Konzentration und Fingerfertigkeit für Telegrafie nachlässt und auch die Hektik beim Arbeiten mit DX-Stationen zu anstrengend ist, werden die digitalen Betriebsarten immer interessanter.

Einen PC hat fast jeder Amateur, eine spezielle Anschluss-Hardware ist nur bedingt erforderlich – sie ist ja meistens gleich, ob bei RTTY, PSK und weiteren digitalen Betriebsarten. Man benötigt nur die entsprechende Software für die Betriebsart und der PC macht dann einen Teil der Arbeit.

Hier ist alles etwas ruhiger und gemütlicher, man kann auch schöne und interessante Stationen arbeiten. Dies bestätigt auch die Altersstruktur: die über 65jährigen werden immer zahlreicher (hi), das heißt die digitalen Betriebsarten werden immer mehr angenommen und umgesetzt. (Siehe auch [1])

Beispiele elektronischer QSL-Karten für JT65-QSOs

CC BY-SA

Was ist JT65?

Seit etwa 10 Jahren gibt es die hier beschriebene digitale Betriebsart JT65. JT65 ist der Oberbegriff für WSJT, FSK 441, JT65C und JT65-HF. Die Betriebsart wurde von K1JT für Erde-Mond-Erde Funkverbindungen (und Meteor-Troposcatter) für Frequenzen im Gigahertzbereich entwickelt. Sie war auch für Verbindungen mit kleinster Leistung im HF/MF-Bereich gedacht.

WSJT – die Abkürzung steht für Weak Signal Communication (nach [2]) und ist ein von K1JT (Prof. J. H. Taylor – einem Nobelpreisträger) speziell für den UKW-Amateurfunk entwickeltes Programmpaket zur Kommunikation mit Signalen sehr schwacher Feldstärken (erster Entwurf aus dem Jahre 2001), geeignet für PCs mit Windows und auch seit einiger Zeit für Linux Betriebssysteme.

Die Softwarelösungen stehen kostenlos zur Verfügung. (Siehe auch [3]) Die Betriebsart JT65-HF (JT65A) ist innerhalb des Programmpakets WSJT (WEAK SIGNAL K1JT) eingebettet. Es wurde von K1JT weiter entwickelt und der Amateurwelt zugänglich gemacht.

Die Betriebsart JT65-HF ist für Funkamateure von Interesse, die unter speziellen Großstadteinschränkungen wie Sendeleistungslimitierung, Störungen und Antennenrestriktionen zu leiden haben. Sie sollten die Chance haben auch im Kurzwellenbereich DX-QSOs zu „fahren“. Bei der Programmentwicklung war ein besonderer Schwerpunkt, die extrem schwachen Signale, welche weit unterhalb der Hörbarkeitsgrenze liegen, so aufzubereiten, dass diese mit der Leistung moderner PCs sowie der Soundkarte aus dem Rauschen herausgeholt und weiter verarbeitet werden können.

Zuerst wurde diese Betriebsart vorzugsweise für EME-Betrieb ausgerichtet, damit konnten extrem schwache Signale für QSO-Betrieb genutzt werden. Seit 2007 sind die langsamen, frequenzgetasteten Signale dieser Betriebsart auch innerhalb der Digimode-Bereiche auf Kurzwelle zu finden.

Unter optimaler Ausnutzung der digitalen Signalabtastungs- und Filtertechnik mittels PC und Soundkarte sowie einer genialen Signalcodierung können Signale bei zu 28 dB unter dem Rauschen lesbar gemacht werden.

Somit ergibt sich also eine geniale Betriebsart für kleine Leistungen und weniger guten Antennen, auch nutzbar auf der Kurzwelle.

Der Erfolg bei JT65 liegt in der schmalbandigen Signalabtastung begründet. So kommt ein Multi-Frequenz-Umschaltverfahren mit 65 verschiedenen Tönen zur Anwendung.

Darauf aufbauend folgen im 2,7 Hz-Abstand weitere 64 Töne als Träger mit spezifischen Dateninformationen zur Anwendung. Hieraus ergibt sich eine Bandbreite von 175 Hz, wobei die Einzeltöne mit nur wenigen Hertz Bandbreite abgetastet werden.

Das zur Vorgeschichte – für den QSO-orientierten Funkamateur ist aber mehr die Anwendung dieser Betriebsart in der Praxis von Interesse. Wer mehr über die Theorie erfahren möchte findet im Internet viele Hinweise auf weitere interessante Artikel und viele Tipps zur Anwendung der speziellen Software. (Siehe beispielsweise [4])

(cc) BY-SA

Der Einstieg

Da ich selbst – Achim (DL7UKA) – zu dem Kreis der Funkamateure gehöre, die nur eingeschränkt (Antennen, Leistung... vergleiche auch [5]) Funkbetrieb machen können, hat mich diese Betriebsart sehr interessiert.

Nach Rücksprache mit Fred (DL8NBJ), den ich auch für etwas „neues“ begeistern konnte, entschlossen wir uns, die kostenfreie Software aus dem Internet zu herunterzuladen – und einfach mal zu sehen, was diese Betriebsart so kann beziehungsweise was wir daraus machen können.

Zu zweit geht alles etwas besser, denn bei Problemen kann man sich besser austauschen. An Funkbetrieb dachten wir noch nicht. Gewisse Vorkenntnisse hatten wir ja schon, denn wir machten ja schon längere Zeit digitalen Betrieb in PSK, RTTY, SSTV und auch andere Digimodes hatten wir auch schon ausprobiert.

Das Programm JT65-HF Version 1.07 war schnell heruntergeladen ([6]) und auf dem PC installiert. Die JT65-HF-Programmversion 1.07 (aus dem Jahr 2007) ist eine Anpassung der JT65A-Protokolle durch W6CQZ mit dem Schwerpunkt der Anwendung auf den hohen Amateurbändern zur erleichterten „Schwach-Signal-Kommunikation“.

Zum Programm gibt es auch eine etwa 50seitige Beschreibung, die man ebenfalls herunterladen kann. In dieser Beschreibung kann man im Detail alle näheren Informationen erhalten. Deshalb möchte ich auch in diesem Artikel mich nicht weiter mit dem Programm beschäftigen. Die Einrichtung des Programms ist relativ einfach – ich war in etwa einer Stunde empfangsseitig QRV.

Bevor man aber konkret an die QSO-Abwicklung gehen kann, muss man einige grundlegende Informationen haben: Auf Kurzwelle gibt es bestimmte Frequenzen , die für diese Betriebsart zugelassen beziehungsweise bandplankonform oder als Vorzugsfrequenzen üblich sind. Der Sender ist grundsätzlich auf USB einzustellen, die Leistung auf 1 – 50 Watt herunterzugregeln. Der PC und Transceiver sind wie bei PSK oder RTTY miteinander verbunden – es sind keine besondere Anpassungen nötig. Wer bisher PSK oder RTTY mit der Software MMVARI, MixW2, DigiPan, MultPSK, MMTTY oder ähnlichem gearbeitet hat, kann jetzt problemlos auch mit JT65-HF arbeiten.

Beispiele elektronischer QSL-Karten für JT65-QSOs

CC BY-SA

Grundlagen

Folgende Frequenzen – jeweils die „dial frequency“ (Frequenz des unterdrückten Trägers) in USB – sind für JT65 vorgesehen:

1,838 MHz, 3,676 MHz, 7,076 MHz, 10,139 MHz, 14,076 MHz,
18,102 MHz, 21,076 MHz, 24,920 MHz, 28,076 MHz

Stellt man seinen Empfänger auf diese Frequenzen ein, so erkennt man schon an den „eigenartigen Tönen“, ob Stationen QRV sind.

Jetzt noch einige Hinweise zu den QSOs: Ein QSO besteht aus 3 Durchgängen, pro Durchgang können nur maximal 13 Zeichen gesendet werden. Ein Durchgang dauert immer 1 Minute. Somit dauert ein QSO immer 6 – 7 Minuten, wenn alles problemlos abläuft.

Eine Besonderheit ist deshalb noch notwendig – die Zeitsynchronisation. Die eigene Zeit und die Zeit der Gegenstation müssen sekundengenau übereinstimmen, das heißt alle Stationen empfangen und senden im gleichen Zeitraster.

Zur automatischen sekundengenauen Einstellung meiner Rechneruhr über das SNTP-Protokoll („simple NTP“; zum NTP-Protokoll siehe Anmerkung [7] im Artikel „WSPR-Mobilbetrieb mit Raspberry Pi“ auf Seite 29) verwende ich das im Internet für private Nutzung frei verfügbare Programm „DIMENSION 4v5.0“ ([7]).

Ablauf einer Verbindung

So jetzt zum QSO – wie schon erwähnt, besteht es aus 3 – 4 Stufen:

1. Durchgang – eigener CQ-Ruf:	`CQ DL7UKA JO50`
will jetzt z. B. DL8NBJ ein QSO beginnen so sendet dieser:	`DL7UKA DL8NBJ JO50`
2. Durchgang, Rapport an die antwortende Station:	`DL8NBJ DL7UKA -10`
Dann sendet der QSO-Partner ebenfalls den Rapport:	`DL7UKA DL8NBJ -12`
3. Durchgang (Bestätigung des erhaltenen Rapports)	`DL8NBJ DL7UKA RRR`
Verabschiedung:	`DL7UKA DL8NBJ 73`
4. Durchgang, anrufende Station verabschiedet sich	`DL8NBJ DL7UKA 73`

Als Rapport dient Signalamplitude relativ zum Gesamtsignal im SSB-Kanal in dB. Dieses Verhältnis wird von der Software automatisch ermittelt, somit sind die Rapporte (im obigen Beispiel -10 db und -12 dB) also echte gemessene Werte.

Auf den ersten Blick sieht das alles etwas kompliziert aus, es ist aber doch recht einfach. Schön ist es, wenn man am Anfang einen QSO-Partner in der Nähe hat, mit dem man vieles ausprobieren kann. Das Programm erklärt vieles selbst und es gibt auch einige Artikel in den einschlägigen Amateurfunk-Zeitschriften oder Internetseiten, über die man sich informieren kann.

Es ist, wenn alles gut eingerichtet ist, ein gemütliches Programm, welches immer mehr genutzt wird (insbesondere von älteren Funkamateuren).

(cc) BY-SA

Eigene Erfahrungen und Erfolge

Ich selbst betreibe diese digitale Betriebsart seit Juni 2011; hin und wieder, wenn auf den Bändern nichts los ist, lausche ich auf die JT65-Signale. Ich habe bis heute so etwa 1500 QSOs mit 100 Ländern geführt, davon sind bereits 90 Länder durch Papier- bzw. elektronische QSLs bestätigt. Die Bestätigung durch eQSL wird sehr häufig angewendet. Da diese Betriebsart auch gut zum Dxen ist, hier einige von mir erreichte DX-Raritäten:

Lesoto (7P8), Zambia (9J2), Trinidad (9Y4), Grönland (OX), Cuba (CO), Bolivien (CP), Honduras (HR), Surinam (PZ), Mexico (XE), China (BY), Indien (VU), Indonesien (YB), Australien (VK), Neuseeland (ZL), Hawaii (KH6), Alaska (KL7), Belau (T8).

In den USA ist diese Betriebsart sehr beliebt. So konnte ich das WAS (Worked all States, also alle US-Staaten) in 2 Jahren arbeiten. Diese Verbindungen sind alle auch bestätigt.

Ich würde mich freuen, wenn ich mit diesem Artikel bei einigen OMs die Neugierde geweckt habe, es auch einmal in JT65 zu versuchen.

Viel Spaß und good DX wünscht Achim (DL7UKA).

Diplom-Beispiele

Text und Bilder: DL7UKA

Quellen und weiterführende Informationen:

[1] JT65 on HF spotting system; URL: http://jt65.w6cqz.org/receptions.html

[2] Artikel„WSJT“ in Wikipedia, Die freie Enzyklopädie.
Bearbeitungsstand: 20. Januar 2014, 12:46 UTC.
URL: http://de.wikipedia.org/w/index.php?title=WSJT&oldid=126674367
(Abgerufen: 24. Februar 2014, 17:22 UTC)

[3] WSJT Homepage von Joe Taylor (K1JT); URL: http://www.physics.princeton.edu/pulsar/K1JT/

[4] JT65-Signale auf Kurzwelle; Eike Barthels (DM3ML);
URL: http://dl0tud.tu-dresden.de/~dm3ml/Artikel/JT65_auf_Kurzwelle.pdf
(Abgerufen: 24. Februar 2014, 17:22 UTC)

[5] DL7UKA - Ein bewegtes Amateurfunkleben; CQ Kronach Nr. 16/2010; Seiten 30ff;
Downloadadresse: http://www.darc.de/uploads/media/cqkc16.web.pdf

[6] Downloadseite für JT65-HF; URL: http://jt65-hf.com/downloads/

[7] SNTP-Zeitsynchronisations-Software „Dimension 4“; URL: http://www.thinkman.com/dimension4/

(cc) BY-SA

Selbstbauprojekt 50 MHz-Transverter von DL9NDK

Beschrieben wird ein 6 m-Transverter, der das Frequenzband 144 MHz bis 146 MHz nach 50 MHz bis 52 MHz linear umsetzt. Trotz optimierter Ausführung mit entsprechenden Filtern ist die Schaltung übersichtlich und relativ nachbausicher. Auf Wunsch wird sie natürlich zur Verfügung gestellt. In einschlägigen Veröffentlichungen trifft man relativ viele Bauanleitungen für das genannte Gerät an. Mir lief auf dem Flohmarkt in Neumarkt im letzten Jahr eine teilweise bestückte Platine von OE9PMJ mit allen Unterlagen über den Weg. Das Vervollständigen und die Inbetriebnahme bereiteten kleine Probleme.

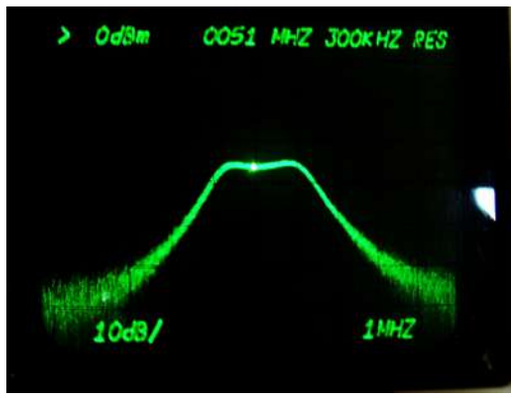
Der fertige 50 MHz-Transverter

- **Empfänger:** HF-Vorstufe DUAL-GATE-MOS-FET BF 960, 3-Kreis-Filter, Mischer IE 500, 145 MHz-Verstärker J310, Empfindlichkeit ca. 0,5 µV für 15 dB S/N.
- **Sender:** Mischer IE 500, 2-Kreis-Filter, DUAL-GATE-MOS-FET BF 960 als geregelter Vorverstärker, BFR 96 als „Endstufe“ für etwa 400 mW Ausgangsleistung.
- **Oszillator:** 94 MHz-Quarzoszillator mit BF 256, Bufferstufe BFW 92, Tiefpass zu den beiden Mischern.

Rückwand mit Anschlüssen

CC BY-SA

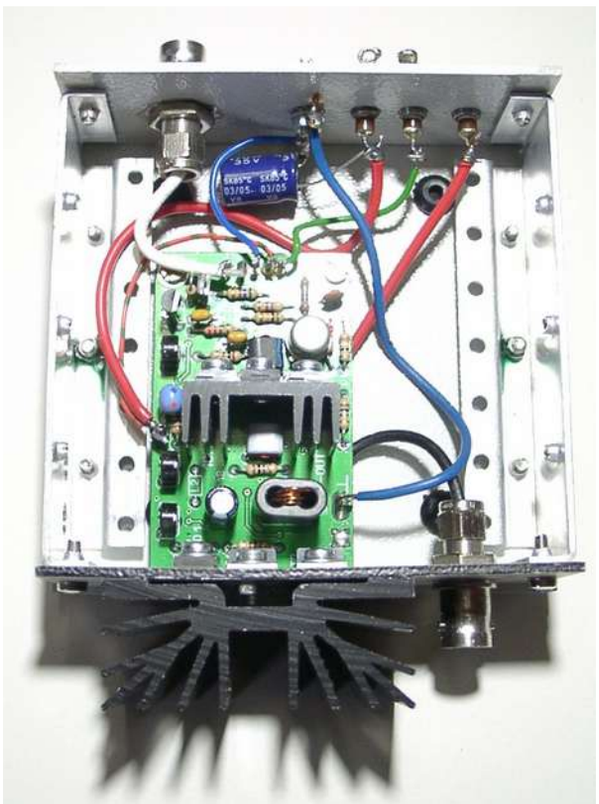
Die Platine des Transverters, oben RX-, unten TX-Zug, in der Mitte der Oszillatorzug



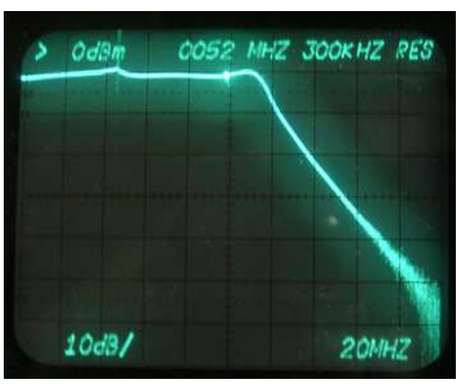
Gewobbelte Durchlasskurve des Eingangsteils

Nachdem die Transverterplatine so gut lief, hielt ich Ausschau nach einer kleinen 50 MHz-Endstufe für welche auch recht viele Schaltungsvorschläge zu finden sind. Im Internet fand ich eine interessante Breitbrandschaltung, welche von DL2AZK ganz hier in unsere Nähe als Bausatz vertrieben wird. Also auf ins ca. 15 km entfernte Neuhaus am Rennweg.

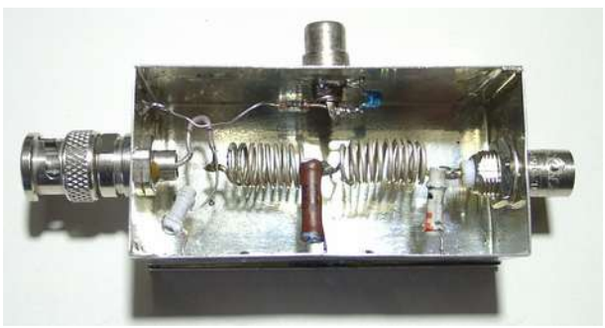
Klaus sicherte zu, dass die PA in den „unteren KW-Bereichen“ ca. 15 W und auch auf 10 m noch etwa 10 W bei 13,8 V bringe. Ein Versuch auf 6 m würde sich lohnen. Nach einem Tag lief die PA, allerdings mit wilden Schwingungen. So war wieder eine Fahrt nach Neuhaus angesagt. Klaus habe so etwas bei noch keiner PA erlebt. Nach zwei Stunden „Forschungsarbeit“ war das Geheimnis gelüftet. Das kleine Gehäuse, das mir Klaus zum Einbau empfahl, isolierte den Kühlkörper der PA-Transistoren von der eigentlichen Masse. Nach Entfernen der Gehäuselackierung arbeitete die PA außergewöhnlich stabil.



10 W-Linearendstufe (1,5 MHz bis 50 MHz)

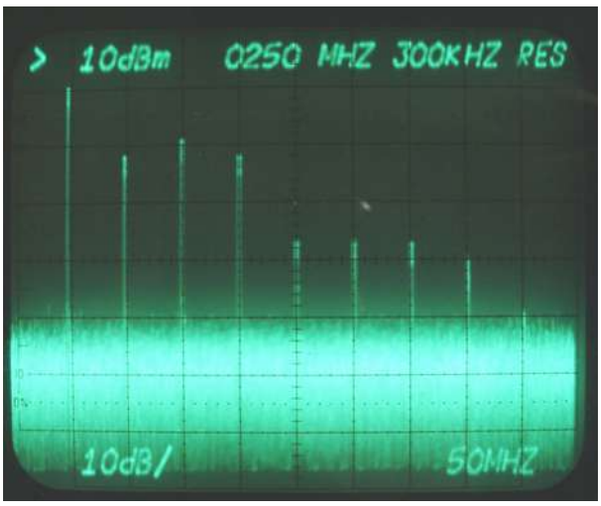


Gewobbelte Durchlasskurve des Tiefpasses.

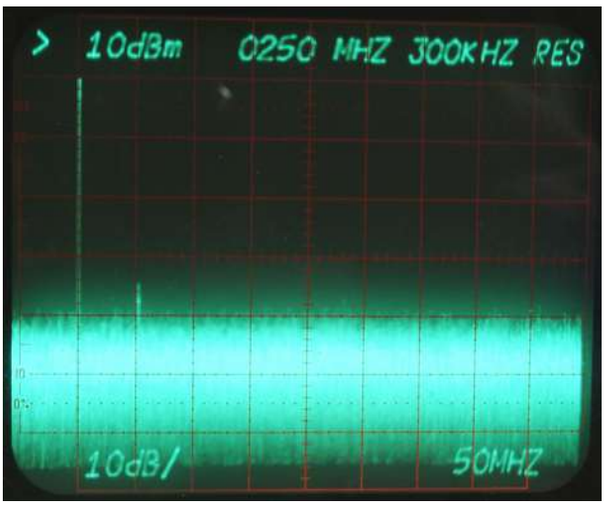


Der Tiefpass

CC BY-SA



Spektrum des Ausgangssignals vor dem Tiefpass. Der „Lattenzaun“ der unzureichend gedämpften Oberwellen ist gut zu erkennen.



Mit nachgeschalteten Tiefpass ist bei der gewählten Darstellung jetzt nur noch eine Oberwelle bei 100 MHz zu sehen.

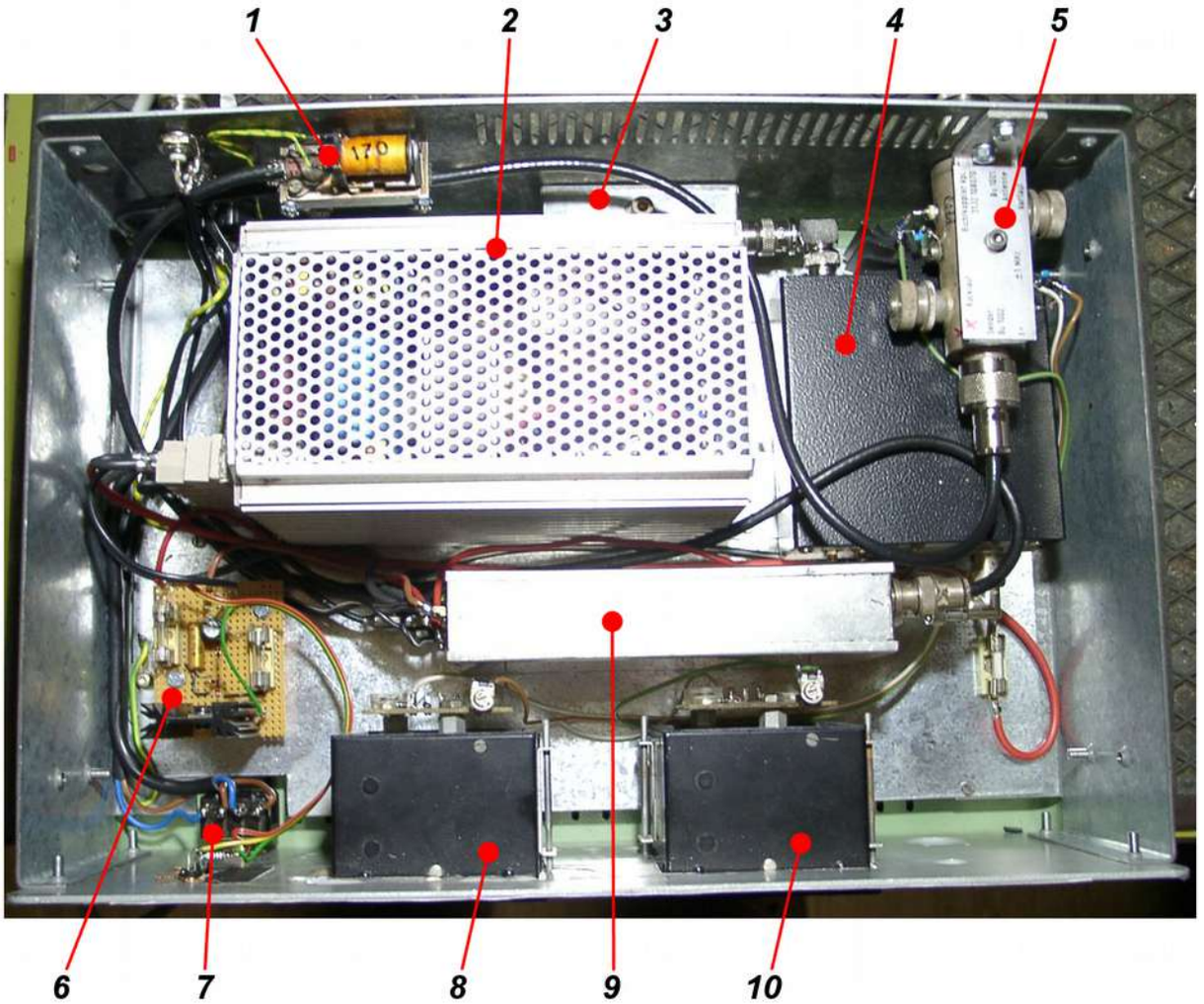
- **Leistungsendstufe:** Vorstufe SF 128 E (normaler „DDR“ Kleinleisungs-HF-Transistor), Treiber 2 x 2 SC 1967 (Si npn), Endstufe 2 x RD15HHF1 (n-Kanal Isolierschicht-FET, selbstsperrend). Nach dem diese Transistoren 30 V DS-Spannung aushalten und die PA bei 13 V nur 5 W erzeugte, wurde die Drainspannung der PA auf 24 V erhöht. Jetzt gab die PA auch bei 50 MHz 10 W an 50 Ohm ab. Die Stromaufnahme beträgt ca. 2,5 A.

- **Tiefpass:** Wird der Sender ohne weitere Selektionsmittel betrieben, so können die postalischen Vorschriften nicht eingehalten werden. Die erste Oberwelle ist lediglich um 12 dB und die zweite (150 MHz) um 10 dB unterdrückt. Bis 350 MHz beträgt die Oberwellendämpfung ca. 30 dB, bei 400 MHz sind es etwa 40 dB. Diese Werte verbessern sich nach Einbau eines Tiefpasses erheblich, wie die Schirmbilder des Spektrumanalysators zeigen: Die erste Oberwelle (100 MHz) ist nun um 45 dB und die zweite Oberwelle (150 MHz) ist mit über 50 dB im Rauschfloor schon nicht mehr sichtbar (die linke Linie entspricht jeweils der Grundwelle 50 MHz; horizontal: 50 MHz/Kästchen; vertikal 10 dB/Kästchen).

- **Stromversorgung:** Weil nur Teile aus der Bastelkiste (außer der PA) verarbeitet werden sollten, stach ein altes 24 V/5 A Philips-Industrie-Schaltnetzteil ins Auge. Diese Spannung stabilisierte ich mit einem normalen 1 A-Regler auf 13,8 V für die übrigen Stufen.

- **Wattmeter:** Eigentlich wollte ich nur ein kleines Messinstrument einbauen, welches bei anliegender Sendeleistung „etwas zappelt“. Da fiel mir ein Geburtstagsgeschenk von Albert (DG5NN) ein. Er hatte mir vor etlichen Jahren einmal eine wunderschöne Messeinrichtung aus einer alten B-Netz-Basisistation (Autotelefon im 2 m – Band) mit gebracht. Diese besteht aus einem toll gearbeiteten Richtkoppler und zwei passenden, in Watt geeichten, Messinstrumenten. Die Eichung für 50 MHz stellte kein Problem dar. Nun verfügt der Transverter über zwei geeichte 30 Watt Messinstrumente für Vor- und Rücklaufleistung.

CC BY-SA



Die Einzelkomponenten im Gehäuse verbaut:

1: Antennenrelais
2: 24 V- Schaltnetzteil
3: 50 MHz-Tiefpass
4: 10 W-Leistungsendstufe
5: Richtkoppler
6: 13,8 V-Spannungsregler
7: Netzschalter
8: Wattmeter: Vorlauf
9: Transverter 50/144 MHz
10: Wattmeter: Rücklauf

CC BY-SA

Erste Betriebserfahrungen

Um den Betrieb aufnehmen zu können erwarb ich eine 5-Element Konni-Antenne. Sie soll 6 dB Gewinn haben. Die Antenne funktioniert derzeit nicht zufriedenstellend. Ob Montage- oder Antennenfehler vorliegen kann erst geklärt werden, wenn die Antenne im Frühjahr noch einmal vom Mast genommen und überprüft wird.

Trotzdem konnten in einer Woche im tiefen Tettau-Tal sechs Länder gearbeitet werden. Die größte Entfernung war Griechenland. Danach ruhte die Schaltung bis zum Einbau vor einigen Tagen in ein Gehäuse.

Unmittelbar nach der Fertigstellung des Versuchsaufbaus lief gerade die 6 m-Band-Hochsaison im Juni. Bisher konnte ich nur die Nürnberger Bake DF6ANN mit S4 aufnehmen (Rundstrahler, 2 W, 50,083 MHz). Der 6 m-Betrieb muss jetzt nur noch angezeigt, nicht mehr genehmigt werden (nur bei Klasse A möglich). Die effektive Strahlungsleistung darf 25 W nicht überschreiten. D. h. bei einer 6 dB-Antenne dürfte man – ohne Kabelverlust – nur 6,25 W einspeisen. Sendebetrieb ist in DL momentan nur zwischen 50,03 und 51 MHz erlaubt. Bei Bandöffnung ist die Hauptaktivität (CW; SSB) zwischen 50,1 und 50,3 MHz.

Da Bandöffnungen oft sehr zeitlich begrenzt auftreten – eigentlich nur im Hochsommer - existiert im 6 m-Band eine andere Betriebsabwicklung als allgemein üblich (kurz, nur Austausch von Rufzeichen, Rapport, QTH-Kenner). Wegen dieser Ausbreitungsbedingungen (einmal wie KW, einmal wie UKW) wird der 50 MHz-Bereich auch Magic Band genannt. Lassen wir den Sommer kommen!

Text und Fotos: DL9NDK

CC BY-SA

WSPR-Mobilbetrieb mit Raspberry Pi

Das Übertragungsverfahren WSPR (**W**eak **S**ignal **P**ropagation **R**eporter) dient zur Untersuchung der Ausbreitungsbedingungen speziell auf Kurzwelle mit geringer Sendeleistung. (Siehe auch [1], [2].) Die Ergebnisse sind auf einer Internetseite ([3]) einsehbar.

Das verwendete Datenübertragungsprotokoll ermöglicht lediglich die Übermittlung stark eingeschränkter Informationen – ausschließlich Rufzeichen, Sendeleistung in dBm und Maidenhead-Standortkenner (dieser standardmäßig nur vierstellig) – auch noch bei sehr geringem Signal-Rauschabstand. Die Aussendungen haben eine Dauer von etwa zwei Minuten, sind sehr schmalbandig und müssen innerhalb eines kleinen Zeitfensters jeweils kurz nach der vollen geradzahligen Minute beginnen. Wenn Rufzeichenzusätze oder ein sechsstelliger Locator übertragen werden sollen erfolgt eine Aufteilung der Daten auf zwei Sendezyklen. Der Wechsel zwischen Sende- und Empfangsbetrieb erfolgt idealerweise über einen Zufallsgenerator. (Siehe [4])

An der heimischen Station ist WSPR-Betrieb mit einem ausreichend leistungsstarken Computer problemlos, die erreichten Reichweiten sind recht überraschend. Ich wollte jedoch auch im Fahrzeug WSPR betreiben, unter anderen um die Leistungsfähigkeit meiner Mobilantennen (vgl. [5]) beurteilen zu können.

WSPR-Mobilbetrieb - erste Versuche

Zunächst verwendete ich im Auto ein Netbook (Asus Eee PC) mit Soundkarteninterface (vgl. [6]), zur Zeitsynchronisation diente NTP (**N**etwork **T**ime **P**rotocol; ein über Internet verfügbarer Zeitdienst; siehe auch [7]) mittels Mobiltelefon, nachdem Versuche mit Freeware-Programmen zur Zeitsynchronisation über GPS eher ernüchternde Ergebnisse brachten. Aufgrund der zu geringen Taktfrequenz des Computers dauerte die Dekodierung der empfangenen Stationen zwar etwas lange, funktionierte aber überraschenderweise ähnlich zuverlässig wie auf dem Shack-Rechner.

Später setzte ich als Audioquelle an Stelle des Netbooks einen einfachen mp3-Player ein, auf dem ich entsprechende Tondateien gespeichert hatte. Die Tondateien enthielten jeweils einen kompletten Datenübertragungszyklus, bestehend aus zwei Sendezyklen (die Aufteilung war wegen der Suffixerweiterung „/m“ nötig) und entsprechend langen Pausen, die Gesamtlänge der Tondateien war so bemessen, dass ich sie in einer Endlosschleife abspielen konnte und sich ein angemessener Sende-Empfangszyklus ergab. Die Wiedergabe startete ich manuell anhand einer Funkuhr. An Stelle eines kompletten NF-Interfaces verwendete ich ein vereinfachtes Verbindungskabel zum Transceiver, welches nur den NF-Sendezweig (NF-Übertrager und Abschwächer) enthielt. Die Sendertastung erfolgte über VOX.

Sowohl beim Netbook als auch beim mp3-Player musste ich jedoch selbst auf den richtigen Locator achten, also beim Wechseln in ein neues Locatorfeld entweder das Programm neu konfigurieren oder eine andere Tondatei starten.

Nachdem ich zunächst keine „fertige“ Softwarelösung, die mittels GPS-Empfänger den richtigen Locator bestimmt, die Aussendungen entsprechend anpasst und zeitgenau startet, finden konnte, beschloss ich, entweder eine Software für den Einplatinencomputer vom Typ Raspberry Pi anzupassen oder selbst ein entsprechendes Programm zu schreiben.

CC BY-SA

Der Raspberry Pi

Der Raspberry Pi, auf den ich unter anderem durch das Amateurfunkmagazin CQ DL ([8]) aufmerksam wurde, ist ein kompakter (üblicherweise als „kreditkartengroß“ beschriebener) Einplatinencomputer mit einem 700 MHz-ARM-Prozessor.

Die Stromversorgung erfolgt über eine Micro-USB-Buchse. An Stelle einer Festplatte wird eine SD-Karte in den integrierten Kartenleser eingesetzt. Als Grafikausgänge stehen ein Video-Cinch-Anschluss und eine HDMI-Buchse zur Verfügung, die Tonausgabe erfolgt über HDMI oder eine 3,5 mm-Klinkenbuchse. Auf der Platine befindet sich zudem eine GPIO-Schnittstelle (mehrere digitale Ein-/Ausgabe-Pins, die frei programmiert werden können).

Platine des Rasperry Pi Version B

Abbildung: Jwrodgers aus: Wikimedia Commons

Es ist weder ein Mikrofon- noch ein Line-In-Eingang vorhanden. Ebenso fehlt eine Echtzeituhr.

Die von mir verwendete Version B verfügt über 512 MB RAM, zwei USB-Ports sowie eine Ethernet-Schnittstelle. Als Betriebssystem dient ein angepasstes Linux (zunächst Raspbian „wheezy“). Um die Platine zu schützen verwende ich ein handelsübliches, in etwa zigarettenschachtelgroßes Gehäuse.

Verfügbare WSPR-Software für den Raspberry

Sucht man in Google nach „Raspberry WSPR“ erhält man unter anderem mehrere Ergebnisse zur Software „WsprryPi“ ([9]). Hiermit erzeugt der Computer direkt ein Rechteck-WSPR-Signal bis zu 250 MHz an der GPIO-Schnittstelle. Nach ausreichender Filterung (Tiefpass) arbeitet der Raspberry Pi somit als Sender mit etwa 10 mW Ausgangsleistung. Allerdings ist der GPIO-Port empfindlich gegen ESD oder andere Überspannungen von der Antenne. Die nötigen Schutzmaßnahmen und umschaltbaren Bandfilter erschienen mir als zu umständlich, weshalb ich diesen Weg nicht weiter verfolgte.

Ebenso findet man eine leicht nachvollziehbare Anleitung ([10]) zum Compilieren der Original-WSPR-Software mit vollem Funktionsumfang. Ich musste allerdings feststellen, dass die Aussendungen verspätet begannen. Vor der Aussendung zur vollen geradzahligen Minute stieg die Prozessorlast stark an, daher vermute ich, dass der Prozessor für die zeitgerechte Berechnung des Audiosignals zu leistungsschwach ist. Eine Anpassung des Quelltextes an diese Bedingungen gelang mir nicht. (Die Software funktioniert übrigens, wenn man den Raspberry übertaktet.)

Bereits im Verlaufe dieser Versuche installierte ich den bei meinem Programm nötigen GPS-Dämon sowie die Bluetooth-Verwaltungssoftware, koppelte den Bluetooth-GPS-Empfänger an die Schnittstelle */dev/rfcomm0* und ergänzte den GPS-Empfänger in der NTP-Konfiguration, so dass die Systemuhr auch über GPS synchronisiert wurde.

(cc) BY-SA

Entwicklung eines einfachen Python-Programmes

Die Lösung mit dem mp3-Player funktionierte eigentlich recht gut, nur der Start und die Auswahl der richtigen Tondatei sollte automatisch erfolgen. Daher beschloss ich, ein einfaches Programm zu erstellen, welches die Daten eines vorhandenen Bluetooth-GPS-Empfängers auswertet und zeitgerecht die zum Standort passenden Tondateien an ein mp3-Abspielprogramm übergibt.

Mit den von mir bis dahin beherrschten Programmiersprachen erschien mir die Einbindung des GPS-Empfängers als zu kompliziert, deswegen entschied ich mich für die Realisierung erstmals die leicht erlernbare Sprache Python ([11]) zu verwenden, was sich tatsächlich als recht einfach erwies.

Lastenheft und grundsätzlicher Programmablauf

Das zu erstellende Programm sollte folgende Funktionen aufweisen :

- ➔ Auslesen der Zeit und der geographischen Koordinaten aus einem Bluetooth-GPS-Empfänger
- ➔ Bestimmung des Locator-Großfeldes anhand der geographischen Koordinaten
- ➔ Zur vollen geradzahligen Minute mit einer einstellbaren Wahrscheinlichkeit entsprechende Tondatei abspielen
- ➔ Anzeige des aktuellen Locators
- ➔ Falls nötig, Zeitversatz aufgrund begrenzter Verarbeitungsgeschwindigkeit kompensieren

Im Laufe der Programmierung und ersten Tests ergaben sich folgende Ergänzungen:

- ➔ Anzeige der Aktion für den jeweils folgenden 2-Minuten-Zyklus; Zählen und Anzeigen der Anzahl der Empfangszyklen
- ➔ Verhinderung von Dauersendungen wegen nicht-aktualisierter Zeit bei Abriss des GPS-Empfangs
- ➔ Optional beim Wechsel des Locatorfeldes auf jeden Fall im nächsten 2-Minuten-Zyklus Aussendung initiieren

Der kommentierte Quelltext der ersten funktionsfähigen Programmversion ist auf den folgenden Seiten abgedruckt. Bereits dieser Quelltext wurde mehrfach erweitert, um Fehler zu beseitigen oder zusätzliche Funktionen einzufügen. Daher ist er an manchen Stellen etwas umständlich. So habe ich in späteren Versionen einige Variablen eingespart. Der Quelltext der späteren Versionen ist jedoch zu umfangreich für einen Abdruck an dieser Stelle, da ich noch einige zusätzliche Funktionen integriert habe.

Die gemäß der verwendeten Freeware-Lizenz „GNU General Public License V3“ nötige Erklärung ist hier aus Platzgründen als Infokasten abgedruckt und nicht als Kommentar ins Programm eingefügt. Eine Anzeigemöglichkeit für den Lizenztext innerhalb des Programms habe ich nicht vorgesehen, da das Programm nicht interaktiv arbeitet.

(cc) BY-SA

Quelltext

(c) DG1NDE 2013 = This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version. = This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, see <http://www.gnu.org/licenses/>

```
#! /usr/bin/python
# written by DG1NDE, 2013
# License: GPL 3
import os
from gps import *
from time import *
import time
import threading
import random
gpsd=None                                          # Globale Variable setzen
os.system('sudo gpsd /dev/rfcomm0 -F /var/run/gpsd.sock')      # GPS einbinden
os.system('clear')                                 # Bildschirm erstmal loeschen
class GpsPoller(threading.Thread):
  def __init__(self):
    threading.Thread.__init__(self)
    global gpsd                                            # in scope bringen
    gpsd=gps(mode=WATCH_ENABLE)                            # Infostream starten
    self.current_value=None
    self.running=True
  def run(self):
    global gpsd
    while gpsp.running:
      gpsd.next()
if __name__=='__main__':
  gpsp=GpsPoller()                                         # Erzeuge thread
  try:
    gpsp.start()                                                  # starte
    liste='JKLMNOPQRSTUV'                                # Fuer Locator-Feldnamen
    list2='0123456789'                                   # Fuer Locator-Feldnamen
    wahrscheinlichkeit=20                                # Sendewahrscheinlichkeit
    sende=True                                                    # Sendeerlaubnis
    teil='a'                                    # Zweiteilige Aussendung bei Mobil
    locatoralt='JJ00'                              # Erkennung Locatorfeldwechsel
    sekunde=99        # Variablendefinition - unmoeglicher Wert fuer Fehlersuche
    zaehler=0                              # Zaehler fuer Anzahl der Empfangsperioden
    zufall=0                                                   # Variablendefinition
    sofort=True                                                # Variablendefinition
    print 'Starte'                                                   # Startanzeige
    print wahrscheinlichkeit,'% TX '
    time.sleep(5)                                # Verzoegerung zur GPS-Einbindung
    while True:                                                  # Endlosschleife...
      locator=''                                    # Variable fuer Locator leeren
      if gpsd.fix.mode<=1:                        # Feststellen ob Position gueltig
         os.system('clear')                                      # Bildschirm loeschen
         print 'NO FIX'                           # Hinweis auf ungueltige Position
      if gpsd.fix.mode>1:
         longi0=gpsd.fix.longitude                  # Geografische Laenge auslesen
         latit0=gpsd.fix.latitude                   # Geografische Breite auslesen
         stunde=int(gpsd.utc[11:13])          # Zeit: Stunde als integer auslesen
         minute=int(gpsd.utc[14:16])          # Zeit: Minute als integer auslesen
         sekunde=int(gpsd.utc[17:19])        # Zeit: Sekunde als integer auslesen
```

Das Programm als Gesamtheit unterliegt der GNU GPL V3.

Einzelne isolierte Programmteile und die verwendeten Algorithmen sind aufgrund geringer Schöpfungshöhe gemeinfrei.

CC BY-SA

```
52          longi1=math.trunc(longi0/20)              # Groesztfeld-Laenge auslesen
53          latit1=math.trunc(latit0/10)              # Groesztfeld-Breite auslesen
54          longi2=math.trunc((longi0-math.trunc(longi0/20)*20)/2)      # Kleinfeld
55          latit2=math.trunc(latit0-math.trunc(latit0/10)*10)          # Kleinfeld
56          locator=locator+liste[longi1]+liste[latit1]+list2[longi2]+list2[latit2]
57          if locatoralt!=locator:     #auskommentieren, falls keine sofortige ...
58            sofort=True               # ... Aussendung bei Locatorfeldwechsel ...
59            teil='a'                                              # ... gewuenscht
60          os.system('clear')                                  # Bildschirm loeschen
61          print stunde,':',minute,':',sekunde,'\nLoc:', locator,teil
62          if not sende:
63             print '!TX',locatoralt,'-->'               # Info zu Kontrollzwecken
64             if zufall<=wahrscheinlichkeit or sofort:
65                print '* TX next',zufall                  # Info TX ohne RX-Phase
66          if sende and (zufall<=wahrscheinlichkeit or sofort):
67             grund='<'       # Symbol fuer Aussendung aufgrund Wahrscheinlichkeit
68             if sofort:
69                grund='!'        # Symbol fuer Aussendung wegen Start/Feldwechsel
70             print 'TX next',zufall,grund               # Info zu Kontrollzwecken
71          else:
72             print 'RX:',zaehler,'Z=',zufall            # Info zu Kontrollzwecken
73          if minute%2==0 and sekunde==0 and sende:
74            befehl='mpg321 -g 100 '+locator+teil+'.mp3 &'        # Programmaufruf
75            if zufall<=wahrscheinlichkeit or sofort:       # wenn Zufall oder ...
76              print 'auf Sendung'                    # ... beim Start Info und ...
77              os.system(befehl) # ... externe Soundausgabe im Hintergrund starten
78              time.sleep(65)
79              sende=False         # wenn GPS-Zeit nicht aktualisiert keine Sendung
80              sofort=False        # Sofortaussendung erfolgt, SofortTX ausschalten
81              locatoralt=locator                  # merke Locator der Aussendung
82              zaehler=1            # Zaehler der Empfangsperioden auf Null setzen
83              if teil=='a':                          # Teil-Aussendung umschalten
84                teil='b'
85              else:
86                teil='a'
87            else:
88              zaehler=zaehler+1                    # Empfangsperioden hochzaehlen
89              time.sleep(2)
90            zufall=random.randrange(1,100)                 # Zufallszahl erzeugen
91            if zaehler>200/wahrscheinlichkeit:           # wenn lange kein TX ...
92               zufall=zufall/2                  # ... Wahrscheinlichkeit erhoehen
93            if zaehler>500/wahrscheinlichkeit:      # wenn sehr lange kein TX ...
94               sofort=True                       # ... Sofortaussendung einleiten
95       time.sleep(0.3)                             # Wartezeit zwischen Abfragen
96       if sekunde>56:                                 # Sendung wieder zulassen
97          sende=True
98    except (KeyboardInterrupt, SystemExit):                          # Bei ctrl+c
99      os.system('clear')
100     print '\nBeende...'
101     gpsp.running = False
102     gpsp.join()                                  # Warten, bis thread beendet
103   print 'Erledigt.'
```

CC BY-SA

Erläuterungen

In Python erfolgt die Strukturierung durch die Einrückungen und nicht durch Klammern oder Schlüsselwörter wie bei vielen anderen Programmiersprachen.

Zum Programmstart wird in Zeile 11 zunächst der Bluetooth-GPS-Empfänger eingebunden. Anschließend erfolgt in Zeile 39 die Anzeige der im Programm definierten Sendewahrscheinlichkeit, die anschließende Verzögerung in Zeile 40 dient auch dazu, dem Computer etwas mehr Zeit für die GPS-Einbindung zu geben.

In der Endlosschleife von Zeile 41 bis 97 befindet sich das Hauptprogramm. In den Zeilen 47 bis 51 werden die Daten des GPS-Empfängers ausgelesen und in Variablen abgelegt, das Locator-Großfeld dann in den Zeilen 52 bis 56 ermittelt. Theoretisch könnte man auch noch das Kleinfeld ermitteln. Dies ist jedoch eher sinnlos, da WSPR nicht als APRS-Ersatz missbraucht werden sollte. Außerdem stiege dadurch die Zahl der nötigen Tondateien stark an.

In den Zeilen 60 bis 72 erfolgt die Statusanzeige. Da einige Anforderungen bezüglich der Anzeige sich erst während der Testläufe ergaben, ist in diesem Bereich das Programm etwas umständlich geschrieben.

Jeweils zu Beginn der geradzahligen Minute wird in Zeile 74 die Befehlszeile für den Aufruf des externen Programs *mpg321* zum Abspielen der passenden Tondatei als Hintergrundprozess erzeugt. Die Dateien beginnen – bedingt durch die ursprüngliche Verwendung im mp3-Player – zunächst mit Stille, so dass die eigentliche Aussendung zeitgerecht erfolgt. Wäre die Verarbeitungsgeschwindigkeit des Python-Programms zu gering könnte das Programm *mpg321* auch eine entsprechende Anzahl an Frames zu Beginn überspringen.

Anschließend wird in Zeile 75 ermittelt, ob die Bedingungen für eine Sendeperiode gegeben sind. Falls ja wird in Zeile 77 die Tonausgabe mit dem in Zeile 74 generierten Befehl gestartet. Die Verzögerung in Zeile 78 dient dazu, die Ausgabe des externen Programms noch ablesen zu können. Dies könnte man auch durch Aufruf als Vordergrundprozess erreichen, jedoch bliebe dann weniger Zeit zur Anzeige der Aktion im nächsten Zyklus. In den Zeilen 83 – 86 wird zwischen den beiden Teil-Aussendungen umgeschaltet.

Sind die Bedingungen für eine Sendeperiode nicht gegeben, wird in Zeile 88 der Zähler für die Empfangsperioden hochgezählt. Die Verzögerung in Zeile 89 verhindert mehrfaches Hochzählen, da die Endlosschleife normalerweise während einer Sekunde mehrfach ausgeführt wird.

In Zeile 90 wird nun die Zufallszahl ermittelt, anhand derer entsprechend der eingestellten Wahrscheinlichkeit der nächste Zyklus ein Sende- oder Empfangszyklus wird. Die Zeilen 91 bis 94 sorgen dafür, dass nach einer größeren Anzahl von Empfangszyklen die Wahrscheinlichkeit eines Sendezyklus erhöht wird.

In Zeilen 96 und 97 wird gegen Ende eines 2-Minuten-Zyklus die Sendeerlaubnis wieder erteilt, die zunächst in Zeile 79 bei Beginn einer Aussendung entzogen wurde. Hiermit wird verhindert, dass eine Daueraussendung initiiert wird wenn aufgrund abgerissenen GPS-Empfangs die Uhrzeit bei einer geradzahligen Minute und Null Sekunden stehen bleibt.

Um das Programm mit Strg+C beenden zu können ist ab Zeile 98 ein entsprechender Exception Handler definiert.

(cc) BY-SA

Anwendung im Mobilbetrieb

Der Rechner selbst wird mittels Klettband an geeigneter Stelle befestigt. Im Mobilbetrieb stellte sich zunächst das Problem der Stromversorgung – ein Abschalten der Spannung, ohne den Rechner zuvor herunter gefahren zu haben, kann bekanntermaßen zu Beschädigungen des Dateisystems führen. Deshalb versorgte ich den Rechner zunächst über einen Kfz-Handylader aus einem 8 Ah-Bleigelakku. Inzwischen wird er aber einfach aus dem Zigarettenanzünder gespeist – für den Fall der Fälle habe ich eine Sicherheitskopie der SD-Karte.

Einbausituation bei der ersten Testfahrt: Raspberry Pi am Mitteltunnel, links daneben das NF-Interface, drahtlose Tastatur auf dem Beifahrersitz, Monitor an der Windschutzscheibe

Zur Anzeige im Auto dient ein kleiner TFT-Videomonitor. Bei den ersten Testfahrten habe ich den Monitor mittels eines Saugnapfhalters an der Windschutzscheibe befestigt. Bei einer Gesamtstromaufnahme von Raspberry und Monitor unterhalb von 0,5 A ist auch bei Versorgung durch den Bleigelakku eine ausreichend lange Betriebszeit sichergestellt.

Der Rechner ist so eingestellt, dass er nach Anlegen der Betriebsspannung die grafische Benutzeroberfläche startet, ohne dass ein Passwort eingegeben werden muss. Das Python-Programm startete ich bei den ersten Testes manuell mittels drahtloser Tastatur, später nahm ich es in den Autostart der grafischen Benutzeroberfläche auf.

Statusanzeige auf TFT-Monitor

Das Python-Programm selbst wird in einem Terminal-Fenster (entspricht quasi der „DOS-BOX“ unter Windows) ausgeführt. Die Schriftgrößen dieses Terminal-Fensters sind so groß gewählt dass die Ausgaben des Programms ausreichend groß (bildschirmfüllend) dargestellt werden – nicht unbedingt elegant, aber wirkungsvoll.

Bei der ersten Testfahrt erhielt ich unter anderem folgende Rapporte – es funktioniert also:

Zeit (UTC)	MHz	SNR	Grid	Pwr	Reporter	RGrid	km
2013-11-01 07:58	14.097165	-5	JO50	5 W	R2ZAC	KO81ia	1803
2013-11-01 08:20	14.097116	-23	JO50	5 W	K9AN	EN50wc	7246
2013-11-01 08:38	14.097118	-11	JN59	5 W	LA9JO	JP99gb	2216
2013-11-01 09:00	14.097085	-14	JN59	5 W	VK2AKU	QF56pg	16338
2013-11-01 09:00	14.097104	-8	JN59	5 W	DL8TW	JN59li	14 *
2013-11-01 13:14	14.097094	-9	JO50	5 W	4X1RF	KM72ls	2779
2013-11-01 13:14	14.097099	+2	JO50	5 W	G4INK	IO70ld	1135

* Entfernung Locatorfeldmitte JN59 – JN59li; tatsächliche Entfernung 70 km (JN59mx – JN59li)

Nachteilig ist jedoch, dass man die Tondateien aufwändig auf einem anderen Rechner erzeugen muss – unterwegs ist man auf die abgespeicherten Tondateien angewiesen. (Offensichtlich kann man mit der Kommandozeilenvariante von WSPR entsprechende wav-Dateien erzeugen, dies habe ich allerdings noch nicht getestet.)

CC BY-SA

Mögliche Erweiterungen

Für das Herunterfahren des Rechners mittels eines Tasters, der an die GPIO-Schnittstelle angeschlossen ist, gibt es bereits fertige Lösungen. Es sollte auch möglich sein, LEDs zur Statusanzeige über die GPIO-Schnittstelle anzusteuern oder auch einen Taster zu implementieren, der – unabhängig vom Zufallsgenerator – eine Aussendung im nächsten Zyklus auslöst.

Vielleicht lässt sich auch ein Optokoppler mittels GPIO ansteuern, so dass man nicht auf die VOX des Funkgerätes angewiesen ist. In diesem Fall könnte man auch Tondateien mit sehr unterschiedlichem NF-Pegel ausgeben um so – ohne am Gerät die Sendeleistung zu verändern – Baken mit unterschiedlicher Leistung auszusenden, ohne Gefahr zu laufen, dass die VOX wegen zu geringer Lautstärke nicht anspricht.

Text und Fotos, sofern nicht anders vermerkt: DG1NDE

⇨ Die Abbildung „Platine des Rasperry Pi Version B“ stammt aus:
http://commons.wikimedia.org/wiki/File:RaspberryPi.jpg?uselang=de
Author: Jwrodgers
Diese Datei ist unter der Creative Commons-Lizenz
Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 nicht portiert lizenziert.
Abgerufen: 31. Oktober 2013, 17:44 UTC

Quellen und weiterführende Informationen:

[1] Artikel „Weak Signal Propagation Reporter“ in Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 16. Juni 2013, 06:38 UTC; URL: http://de.wikipedia.org/w/index.php?title=Weak_Signal_Propagation_Reporter&oldid=119599374 (Abgerufen: 30. Oktober 2013, 11:21 UTC)

[2] Wellenflüstern mit WSPR; Eike Barthels (DM3ML); Funkamateur 3/2009 auch verfügbar unter URL: http://dl0tud.tu-dresden.de/~dm3ml/Artikel/WSPR.pdf

[3] WSPRnet – Weak Signal Propagation Reporter Network; URL: http://www.wsprnet.org/

[4] WSPR 2.0 Benutzerhandbuch, Revision November 23, 2009; Joe Taylor (K1JT), Übersetzung von DL3LST: URL: http://physics.princeton.edu/pulsar/K1JT/WSPR_2.0_User_German.pdf

[5] Die Mobilstation von Jürgen (DG1NDE); DG1NDE; CQ Kronach Nr. 17/2013 Downloadadresse: http://www.darc.de/uploads/media/cqkc17.pdf

[6] Artikelserie Einstieg in PSK31, Teil 2; DG1NDE; Funkamateur 8/2008

[7] Artikel „Network Time Protocol“ in Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 27. August 2013, 06:37 UTC. URL: http://de.wikipedia.org/w/index.php?title=Network_Time_Protocol&oldid=121949663 (Abgerufen: 30. Oktober 2013, 12:09 UTC)

[8] Titelthema: „Kleincomputer in Kreditkartengröße – Amateurfunk mit dem Raspberry Pi“ sowie mehrere Beiträge ab Seite 239; CQ DL 4/2013

[9] Bareback LF/MF/HF/VHF WSPR transmitter using a Raspberry Pi URL: https://github.com/threeme3/WsprryPi

[10] RaspberryPi WSPR; URL: http://www.george-smart.co.uk/wiki/RaspberryPi_WSPR sowie Compiling WSPR; URL: http://www.george-smart.co.uk/wiki/Compiling_WSPR

[11] Python Programming Language – Official Website; URL: http://www.python.org/

(cc) BY-SA

Kurzfassung der Creative Commons Lizenz CC BY-SA 3.0

creative commons

Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland (CC BY-SA 3.0)

Diese „Commons Deed" ist lediglich eine vereinfachte Zusammenfassung des **rechtsverbindlichen Lizenzvertrages** in allgemeinverständlicher Sprache.

Die "Commons Deed" ist kein Lizenzvertrag. Sie ist lediglich ein Referenztext, der den zugrundeliegenden Lizenzvertrag übersichtlich und in allgemeinverständlicher Sprache aber auch stark vereinfacht wiedergibt. Die Deed selbst entfaltet keine juristische Wirkung und erscheint im eigentlichen Lizenzvertrag nicht. Creative Commons ist keine Rechtsanwaltsgesellschaft und leistet keine Rechtsberatung. Die Weitergabe und Verlinkung des Commons Deeds führt zu keinem Mandatsverhältnis.

Sie dürfen:

das Werk bzw. den Inhalt vervielfältigen, verbreiten und öffentlich zugänglich machen
Abwandlungen und Bearbeitungen des Werkes bzw. Inhaltes anfertigen
das Werk kommerziell nutzen

Zu den folgenden Bedingungen:

Namensnennung — Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen.

Weitergabe unter gleichen Bedingungen — Wenn Sie das lizenzierte Werk bzw. den lizenzierten Inhalt bearbeiten oder in anderer Weise erkennbar als Grundlage für eigenes Schaffen verwenden, dürfen Sie die daraufhin neu entstandenen Werke bzw. Inhalte nur unter Verwendung von Lizenzbedingungen weitergeben, die mit denen dieses Lizenzvertrages identisch oder vergleichbar sind.

Wobei gilt:

Verzichtserklärung — Jede der vorgenannten Bedingungen kann **aufgehoben** werden, sofern Sie die ausdrückliche Einwilligung des Rechteinhabers dazu erhalten.

Public Domain (gemeinfreie oder nicht-schützbare Inhalte) — Soweit das Werk, der Inhalt oder irgendein Teil davon zur **Public Domain** der jeweiligen Rechtsordnung gehört, wird dieser Status von der Lizenz in keiner Weise berührt.

Sonstige Rechte — Die Lizenz hat keinerlei Einfluss auf die folgenden Rechte:

- Die Rechte, die jedermann wegen der Schranken des Urheberrechts oder aufgrund gesetzlicher Erlaubnisse zustehen (in einigen Ländern als grundsätzliche Doktrin des **fair use** etabliert);
- Das Urheberpersönlichkeitsrecht des Rechteinhabers;
- Rechte anderer Personen, entweder am Lizenzgegenstand selber oder bezüglich seiner Verwendung, zum Beispiel Persönlichkeitsrechte abgebildeter Personen.

Hinweis — Im Falle einer Verbreitung müssen Sie anderen alle Lizenzbedingungen mitteilen, die für dieses Werk gelten. Am einfachsten ist es, an entsprechender Stelle einen Link auf die Seite **http://creativecommons.org/licenses/by-sa/3.0/de/** einzubinden.

Verwenden Sie diese Lizenz doch auch für Ihre eigenen Werke bzw. Inhalte.

BY-SA

Für den externen Leser: Wir über uns...

Der **Deutsche Amateur Radio Club e.V.** (DARC) vereinigt aktuell als Dachverband mehr als 40 000 deutsche Funkamateure in über 1000 Ortsverbänden. Er entstand 1951 aus dem Zusammenschluss der vier damaligen Amateurfunkclubs (DARC/BZ, BARC, HRC, WBRC), die sich nach 1945 in den westlichen Besatzungszonen zusammengefunden hatten. Nach der Wiedervereinigung schlossen sich ihm auch die Mitglieder des früheren RSV (Radiosportverband der DDR) an. Korporativ ist der VFDB (Verein der Funkamateure der Deutschen Bundespost) angegliedert. Die Clubzentrale des DARC befindet sich in 34225 Baunatal, Lindenallee 4; der Verein ist im Register des Amtsgerichtes Kassel eingetragen und als gemeinnützig anerkannt.

Im Oktober 1959 wurde der Ortsverband Kronach gegründet. Er betreibt unter anderem die Relaisfunkstelle DB0KCH in Kronach-Gehülz. Als Frequenzen für den lokalen Funkbetrieb werden 145,350 MHz (Direktverkehr, FM), 433,500 MHz (Direktverkehr, FM) sowie 438,725 MHz (Relais DB0KCH, FM) genutzt.

CQ Kronach versorgt einmal jährlich Mitglieder und Freunde unseres Ortsverbandes mit Informationen, Hinweisen und Tipps aus unserer Gegend und versucht, auch nicht technisch vorbelasteten Lesern einen Einblick in ein außergewöhnliches Hobby zu geben. Amateurfunk bietet nicht nur die Möglichkeit, drahtlos mit der ganzen Welt in Kontakt zu treten, sondern vor allem den Einstieg zum Verständnis von Natur (Physik) und Technik und viele weitere Möglichkeiten der persönlichen Entfaltung. Wir wünschen uns, möglichst viele mit CQ Kronach zum Nachdenken und zu eigenen Ideen für eine sinnvolle Freizeitbeschäftigung anzuregen.

Weitere Informationen zu Hobby und Verein können bei der Clubzentrale oder auch direkt beim Ortsverband Kronach nachgefragt werden. Selbstverständlich steht Interessenten auch umfangreiches Material im Internet zur Verfügung (→ http://www.darc.de/ oder → http://www.darc.de/distrikte/b/21/) – schauen Sie doch einmal 'rein!

Termine 2014

11.07.14 – 13.07.14 Fieldday auf dem Wachtelberg bei Wildenberg

Haben Sie Fragen zum Hobby Amateurfunk oder möchten uns kennen lernen? Wir freuen uns auf Ihren Besuch!

CC BY-SA

Wir danken folgenden Firmen, die durch Spenden den Druck dieser Ausgabe ermöglicht haben:

WORK MICROWAVE

Analog & Digital RF-Solutions

Die Firma WORK Microwave, mit Sitz in Holzkirchen, entwickelt und produziert seit 1986 High-Tech-Systeme für:

- Satellitenkommunikation
- Radarsysteme
- HF-Sensoren
- Navigationssimulatoren

www.work-microwave.de

Adresse: WORK Microwave GmbH
Raiffeisenstraße 12
83607 Holzkirchen
Telefon: +49 8024 6408 0
E-Mail: sales@work-microwave.de

HEINZGLAS
family-owned since 1622

Technologie.
Internationalität.
Perspektiven!

Interesse?

HEINZ-GLAS
GmbH & Co.KGaA
Glashüttenplatz 1-7
96355 Kleintettau
Tel. +49 9269 77-100
sales@heinz-glas.com

www.heinz-glas.com