

amateurfunk

MEHR ALS
3GB SOFTWARE DOWNLOAD

2022

FUNKAMATEUR

KOSTENLOS

Software für den FUNKAMATEUR

Neuigkeiten,
Updates,
Lesestoff,
Software zu
Amateurfunk,
Elektronik, PC, Internet und mehr...



- Mehr Leistung mit dem **ESP32**
- Online-Rechner-Verzeichnis
- **DX-Lab Spot Collector**

- GETESTET**
- Automatiktuner ATU100 1,8-55 MHz
 - Global Overlay Mapper
 - Spektrumanalysator RF-Explorer WSUB1G-Plus

- BASTELN**
- Arduino: Schalten mit dem Browser
 - Logikanalysator mit Arduino
 - Relais schalten ohne Programmieren
 - Digital-Analog-Wandler mit Software
 - Funktionsgenerator mit ESP32

- WISSEN**
- Elektrische Felder simulieren
 - Höhenprofile online erstellen
 - Monte-Carlo-Analyse mit MicroCap12
 - Smith-Diagramm angewandt mit SimSmith
 - Empfangen: Zahlen- und Nummersender



VORGESTELLT

- Morsespiel von DC1TH
- Timer **TPL5110** für Batteriegeräte

01
4 194026 814904

2198-9656

Tecsun
PL-990x
versus
PL-880

HÖRVERGLEICH



Alles rund um Amateurfunk

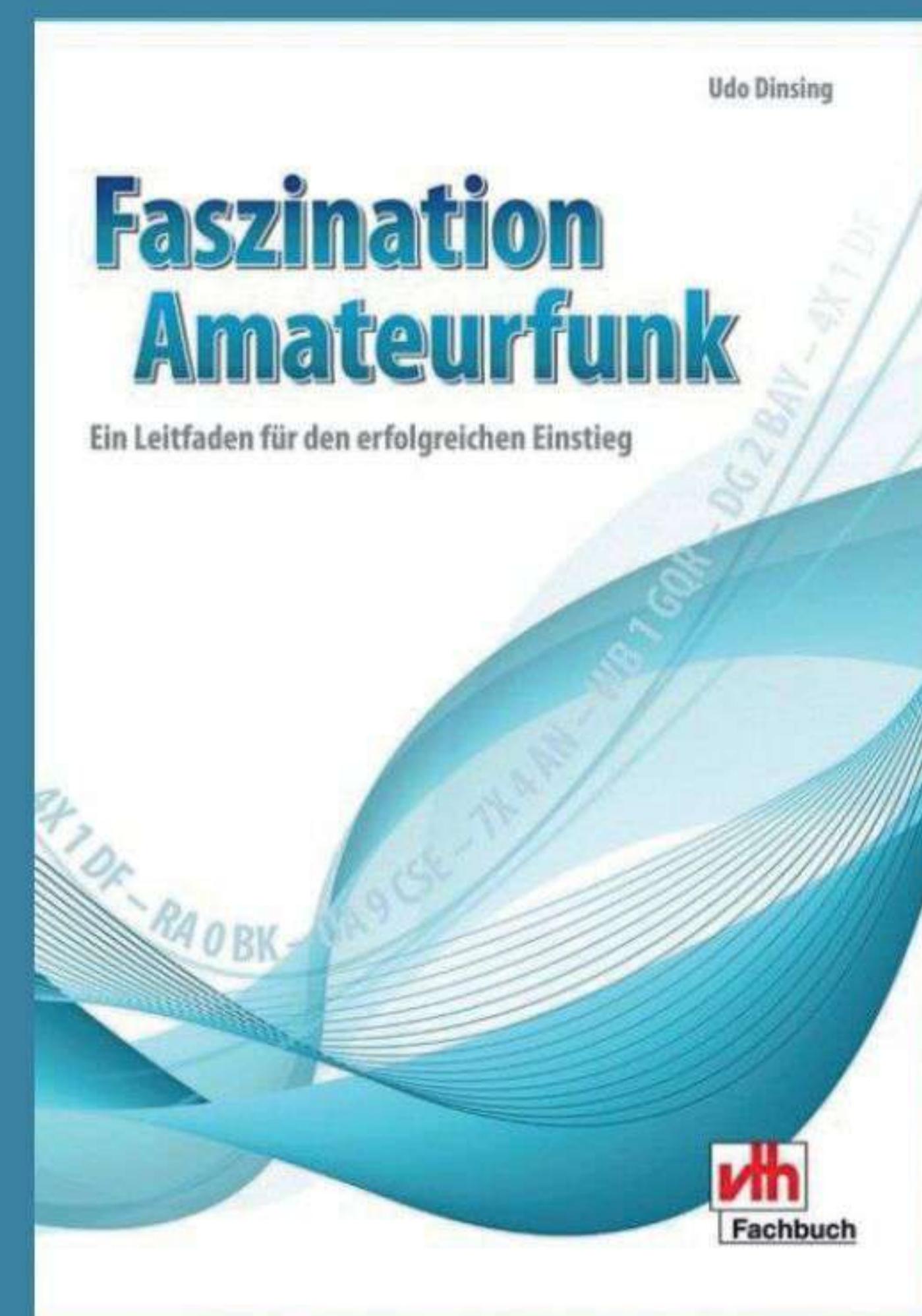


Eckart K.W. Moltrecht, DJ4UF
Amateurfunk-Lehrgang
für das Amateurfunkzeugnis Klasse E

10. neu bearbeitete Auflage

Dieser Lehrgang basiert auf dem Prüfungsfragenkatalog für das Amateurfunkzeugnis Klasse E der Bundesnetzagentur. Alle darin vorkommenden Themen aus den Bereichen Grundlagen der Elektrotechnik, der Elektronik sowie der Sender- und Empfängertechnik, Antennentechnik und Messtechnik einschließlich der Berechnungen für die elektromagnetische Verträglichkeit werden so ausführlich erläutert, dass der Prüfling in die Lage versetzt wird, jede Frage aus dem Fragenkatalog richtig beantworten zu können.

Umfang: 240 Seiten • 300 Abbildungen • Art-Nr: 4110064

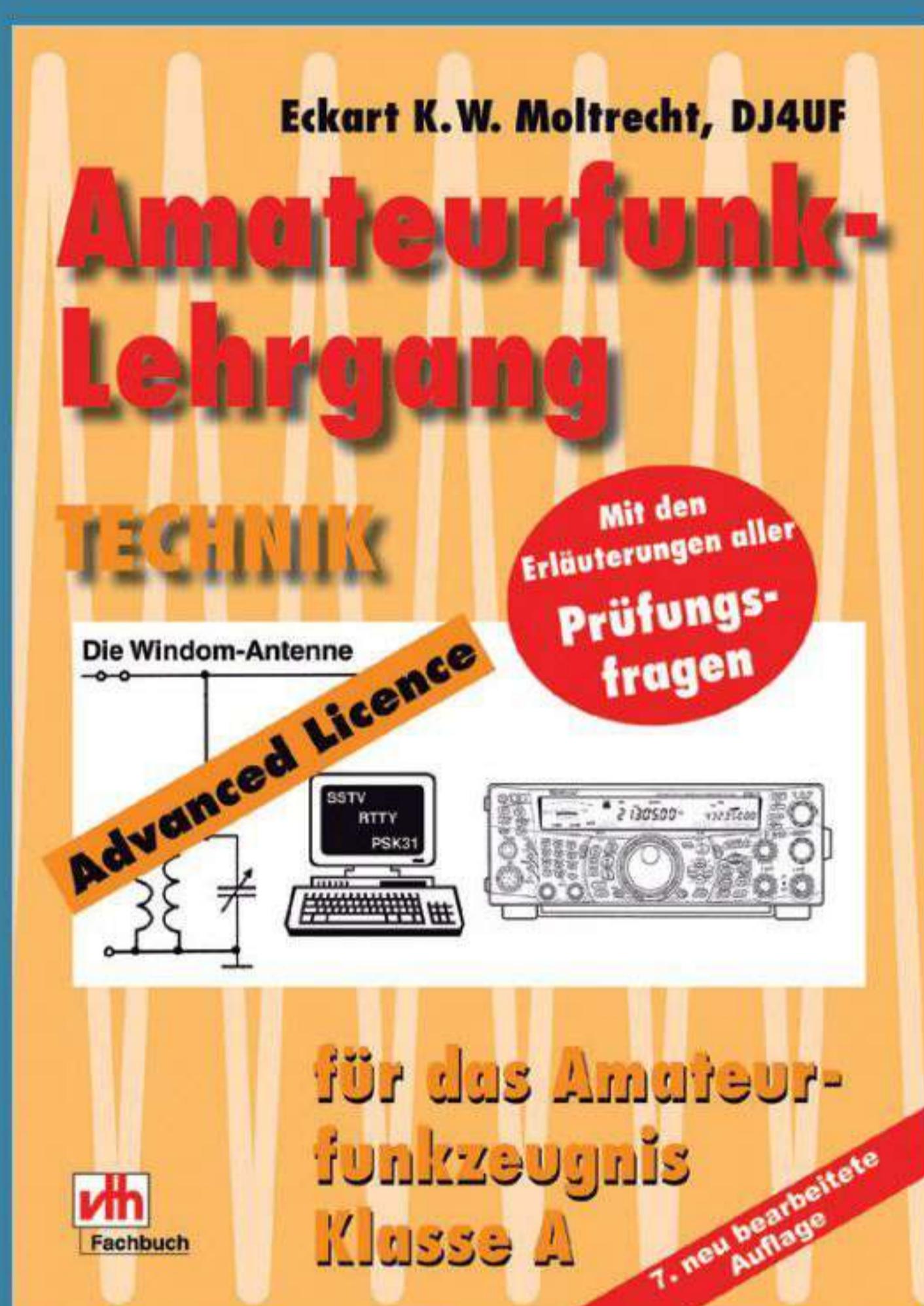


Udo Dinsing
Faszination Amateurfunk

Amateurfunk ist ein spannendes Hobby – doch wie steigt man in dieses faszinierende Thema ein? Udo Dinsing, selbst langjähriger erfolgreicher Amateurfunker, erklärt hier die Grundlagen, wie man zur geeigneten Lizenz kommt, welche technische Ausstattung notwendig und sinnvoll ist und wie man dieses Hobby noch spannender und abwechslungsreicher gestalten kann.

Dieses Buch ist ein Leitfaden für Sie, welcher einen erfolgreichen Einstieg in die Welt des Amateurfunks garantiert.

Umfang: 72 Seiten • ArtNr: 4110161



Eckart K.W. Moltrecht
Amateurfunklehrgang
für das Amateurfunkzeugnis Klasse A Technik
7. überarbeitete Auflage

Alle vorkommenden Themen aus den Bereichen Mathematische Grundlagen, Elektrotechnik, Elektronik sowie Sender- und Empfängertechnik, Übertragungstechnik, Antennentechnik und Messtechnik aus dem Bereich „Technische Kenntnisse“ werden in diesem Band ausführlich erläutert.

Umfang: 304 Seiten • ArtNr: 4110089



Eckart K.W. Moltrecht
Amateurfunklehrgang
Betriebstechnik und Vorschriften
5. überarbeitete und erweiterte Auflage

Dieses Buch ist die ideale Ergänzung zum Amateurfunklehrgang TECHNIK für das Amateurfunkzeugnis Klasse E oder Klasse A. Es sollte parallel zur Technik bearbeitet werden, also etwa eine Lektion Technik und eine Lektion Betriebstechnik/Vorschriften.

Umfang: 160 Seiten • ArtNr: 4110103

Jetzt bestellen!

07221 - 5087-22

07221 - 5087-33

service@vth.de

www.vth.de/shop

vth_modellbauwelt

VTH neue Medien GmbH

VTH & FMT

VTH Verlag

vth

INHALT

Impressum



Software für den Funkamateuer 2022
Bestell-Nr. 300 0105

Verlag für Technik und Handwerk neue Medien GmbH,
Bertha-Benz-Straße 7, 76532 Baden-Baden

Verantwortlicher Redakteur
Eric Scharfenort

Layout
Thomas Schüle

Geschäftsführer
Julia-Sophia Ernst-Hausmann

Anzeigen
Christina Meyhack Tel. 07221 / 5087 -15
Sinem Isbeceran Tel. 07221 / 5087 -90
E-Mail: Anzeigen@vth.de, Fax 07221 / 5087 -33



Verlag

Verlag für Technik und Handwerk neue Medien GmbH
Bertha-Benz-Straße 7, 76532 Baden-Baden
Telefon 0 72 21/5087-0, Telefax 0 72 21/5087-52
Anschrift von Verlag, Redaktion, Anzeigen und allen
Verantwortlichen, soweit dort nicht anders angegeben.

Konten
Grenke Bank AG
IBAN DE45 2013 0400 0060 0368 29
BIC/SWIFT GREBDEH1

Internet: www.vth.de

Vertrieb
MZV Moderner Zeitschriften Vertrieb GmbH & Co. KG
Ohmstraße 1, D-85716 Unterschleißheim
Tel.: 0 89 / 3 19 06-0, Fax: 0 89 / 3 19 06-113

Einzelheft: € 12,80; CH: sfr 25,30; A: € 14,65;
B/NL/L: € 15,10; E/I: € 17,45

Druck

Dierichs Druck & Media GmbH & Co. KG, Kassel

 Amateurfunk wird auf umweltfreundlichem,
chlorfrei gebleichtem Papier gedruckt.

Für unverlangt eingesandte Beiträge kann keine Ver-
antwortung übernommen werden. Mit Übergabe der
Manuskripte und Abbildungen an den Verlag versichert
der Verfasser, dass es sich um Erstveröffentlichungen
handelt und dass keine anderweitigen Copy- oder
Verlagsverpflichtungen vorliegen. Mit der Annahme
von Aufsätzen einschließlich Bauplänen, Zeichnungen
und Bildern wird das Recht erworben, diese auch in
anderen Druckerzeugnissen zu vervielfältigen.

Eine Haftung für die Richtigkeit der Angaben kann
trotz sorgfältiger Prüfung nicht übernommen werden.
Eventuell bestehende Schutzrechte auf Produkte oder
Produktnamen sind in den einzelnen Beiträgen nicht
zwingend erwähnt. Bei Erwerb, Errichtung und Betrieb
von Sende- und Empfangsanlagen sind die gesetz-
lichen und postalischen Bestimmungen zu beachten.
Namentlich gekennzeichnete Beiträge geben nicht in
jedem Fall die Meinung der Redaktion wieder.

© 2021 by Verlag für Technik und Handwerk
neue Medien GmbH, Baden-Baden

Nachdruck von Artikeln oder Teilen daraus,
Abbildungen und Bauplänen, Vervielfältigung und
Verbreitung durch jedes Medium, sind nur
mit ausdrücklicher, schriftlicher Genehmigung des
Verlages erlaubt.

Bildnachweis Titelbilder:
© blackzheep / 123RF.com
© terovesalainen / 123RF.com
Krakenimages.com – stock.adobe.com

Downloads statt DVD

Die virtuelle Heft-DVD im VTH-Shop4

RF-Explorer WSUB1G-Plus

Spektren entdecken6

NodeMCU mit 32-Bit-Controller

Wenn ein Arduino nicht reicht10

Arduino als digitaler Schnüffler

Logikanalysator für das Hobby14

FT8 und FT4

Tipps zum erfolgreichen Betrieb16

Leistungsanpassung für Sender und Antenne

SimSmith für Dummies20

Tecsun PL-990x vs. PL-880

Ein Hörvergleich22

Schalten und Walten per LAN

Arduino UNO und Ethernet Shield24

USB-RELAYS

Schalten mit PC, aber ohne Löten
und Programmieren29

MicroCAP12

Monte-Carlo-Analyse30

Ihre Logs auf der Karte

Global Overlay Mapper34

Spionagesendungen auf der Kurzwelle

Nummernsender38

Das Morsespiel

Lernspaß garantiert39

„Kurzwellen-Drahtantennen“ als PDF-Buch

Bonus-Lesestoff41

Höhenprofile erstellen

Mehrere Lösungen online42

Flüssiger schreiben

TIPP für Android44

Lesevergnügen aus der Vergangenheit

Old Radio Books45

Nano Timer TPL5110

Der Stromspar-Timer46

DX-Lab Spot Collector

Einblick in die Bänder48

Sinus, Rechteck, Dreieck

Funktionsgenerator mit ESP3249

N7DCC-ATU-100

Angepasster China-Nachbau51

Digital-Analogwandler mit Software

Ein Timer-Tutorial mit Atmel ATega32852

Feldlinien simulieren und messen

Ladungen und Felder57

Kleines Verzeichnis der Elektronik-Rechner

Programmidee umgesetzt58

Was soll man mit einer DVD anfangen, wenn der Laptop oder der Notebook-PC kein DVD-Laufwerk besitzt? Man kann einen Freund fragen, ob er die DVD auf einen USB-Stick umkopiert. Das ist zwar eine Möglichkeit, doch nicht jeder Leser hat jemanden in der Nähe, der das erledigen kann. Andererseits ist es nur noch selten ein Problem, drei oder vier Dateien von der Größe eines Gigabytes aus dem Internet zu laden. Das LTE-Netz ist inzwischen auch auf dem Lande gut ausgebaut und bietet recht flotten Zugang zum weltweiten Netz. Beachten Sie bitte: Bei LTE ist die Geschwindigkeit oft von der Tageszeit abhängig, da sich mehrere Teilnehmer die Bandbreite eines LTE-Zugangs teilen.

Zum Download der virtuellen Heft-DVD reicht jedoch schon ein durchschnittlicher DSL-Anschluss aus. Wer mit seinen Internetzugang in der Lage ist, ein Video auf den heimischen Rechner zu laden oder es halbwegs ruckelfrei anzuschauen, wird mit dem Download der Dateien sicher gut fahren.

So geht das mit dem Download

Der Download aus dem VTH-Shop funktioniert so: Im folgenden Text geben wir Ihnen eine Internet-Adresse (URL) bekannt, auf der die Dateien direkt zum Laden auf Ihren PC bereitstehen. Einfacher geht es nicht! Sie müssen sich nicht mehr – wie im letzten Jahr – in den VTH-Shop anmelden, keine Bestellung ausfüllen oder einen Gutscheincode eingeben. Sie rufen die weiter unten genannte Webseite so oft auf, wie Sie möchten und laden die Dateien, wie es nötigt ist. Sie müssen keine Sorge haben, dass eine Datei nach dem Download Fehler aufweist oder gar unbrauchbar wird. Falls ein Download dennoch unerwartet abbricht oder Fehler passieren, laden Sie die betreffende Datei noch einmal auf Ihren PC.

Zum Download werden über die weiter unten genannte Webadresse folgende Dateien angeboten:

Afu_DVD_2022_1.zip
Afu_DVD_2022_2.zip
Afu_DVD_2022_3.zip
Afu_DVD_2022_4.zip ...

Downloads statt DVD

Die virtuelle Heft-DVD im VTH-Shop

Wie bereits im letzten Jahr haben sich Verlag und Redaktion dazu entschlossen, die beliebte Heft-DVD zum Download anzubieten, sozusagen als virtuelle DVD. Die Gründe sind eine bessere Verfügbarkeit des schnellen Internets in Stadt und Land und die Tatsache, dass viele Laptops aufgrund der flachen Bauweise kein DVD-Laufwerk mehr besitzen können. Gegenüber dem letzten Heft wurde der Download der Dateien, die zusammen die virtuelle DVD ergeben, für Sie als Leser wesentlich vereinfacht.

Es werden etwa zehn Dateien sein, die genaue Zahl steht zur Zeit, in der dieser Beitrag entsteht, noch nicht endgültig fest. Jede gepackte Zip-Datei wird etwa 300 MByte betragen. Nach unserer Erfahrung sind Dateien dieser Größe auch bei langsamer Internetverbindung erfolgreich übertragbar. Und nun starten Sie den Download Ihrer Dateien. Die Schritt-für-Schritt-Anleitung macht es Ihnen einfach:

Schritt 1:

Rufen Sie in Ihrem Browser die Webseite <https://www.vth.de/amateurfunk2022> auf. Auf dieser Webseite sehen Sie bereits die Liste der zu ladenden Dateien (Bild 1).

AMATEURFUNK 2022 - DOWNLOADS

Neuigkeiten, Updates, Lesestoff, Software zu Amateurfunk, Elektronik, PC, Internet und mehr..

Downloadpakete

(Der Download startet nach dem Klick auf das Downloadpaket automatisch)

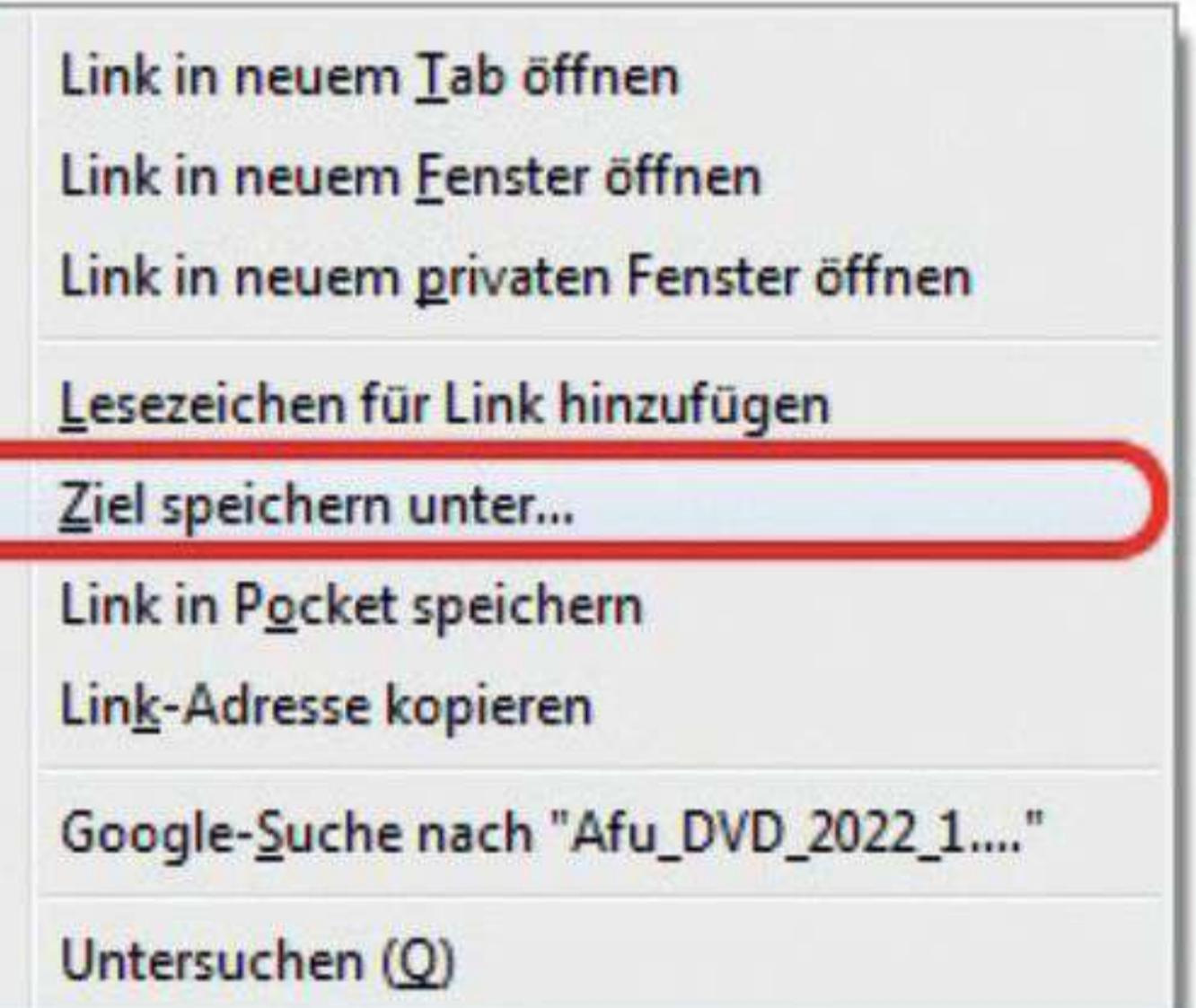
- Afu_DVD_2022_1.zip
- Afu_DVD_2022_2.zip
- Afu_DVD_2022_3.zip
- Afu_DVD_2022_4.zip
- Afu_DVD_2022_5.zip
- Afu_DVD_2022_6.zip
- Afu_DVD_2022_7.zip
- Afu_DVD_2022_8.zip
- Afu_DVD_2022_9.zip
- Afu_DVD_2022_10.zip



Downloadpakete

(Der Download startet nach dem Klick auf das Downloadpaket automatisch)

-
- Afu_DVD_2022_1.zip
 - Afu_DVD_2022_2.zip
 - Afu_DVD_2022_3.zip
 - Afu_DVD_2022_4.zip
 - Afu_DVD_2022_5.zip
 - Afu_DVD_2022_6.zip
 - Afu_DVD_2022_7.zip
 - Afu_DVD_2022_8.zip
 - Afu_DVD_2022_9.zip
 - Afu_DVD_2022_10.zip



▲ Bild 2: Ein Rechtsklick mit der Maus auf die Zip-Datei öffnet das Kontextmenü. Wählen Sie die Option Ziel speichern unter. (Sollten Sie die links Maustaste benutzen, speichert Windows alle Dateien auf Ihrem PC unter Downloads. Sie müssen Sie dann später in ein leeres Verzeichnis kopieren und dort auspacken.)

Sie die Möglichkeit, das Verzeichnis zu bestimmen, in dem die Dateien auf Ihrer Festplatte abgelegt werden. Empfohlen ist z. B. C:\AfuSoftwareDVD2022. Legen Sie mit dem Windows Explorer zuerst dieses Verzeichnis, auch als Ordner bezeichnet, an. Nach dem Mausklick auf Ziel speichern unter erscheint ein Dialog nach Bild 3. Dort legen Sie das Zielverzeichnis fest (roter Rahmen oben). Schließen Sie den Dialog mit einem Klick auf Speichern.

◀ Bild 1: Die Webseite mit der Liste der zum Download bereitstehenden Dateien. Alle gemeinsam bilden die virtuelle DVD.

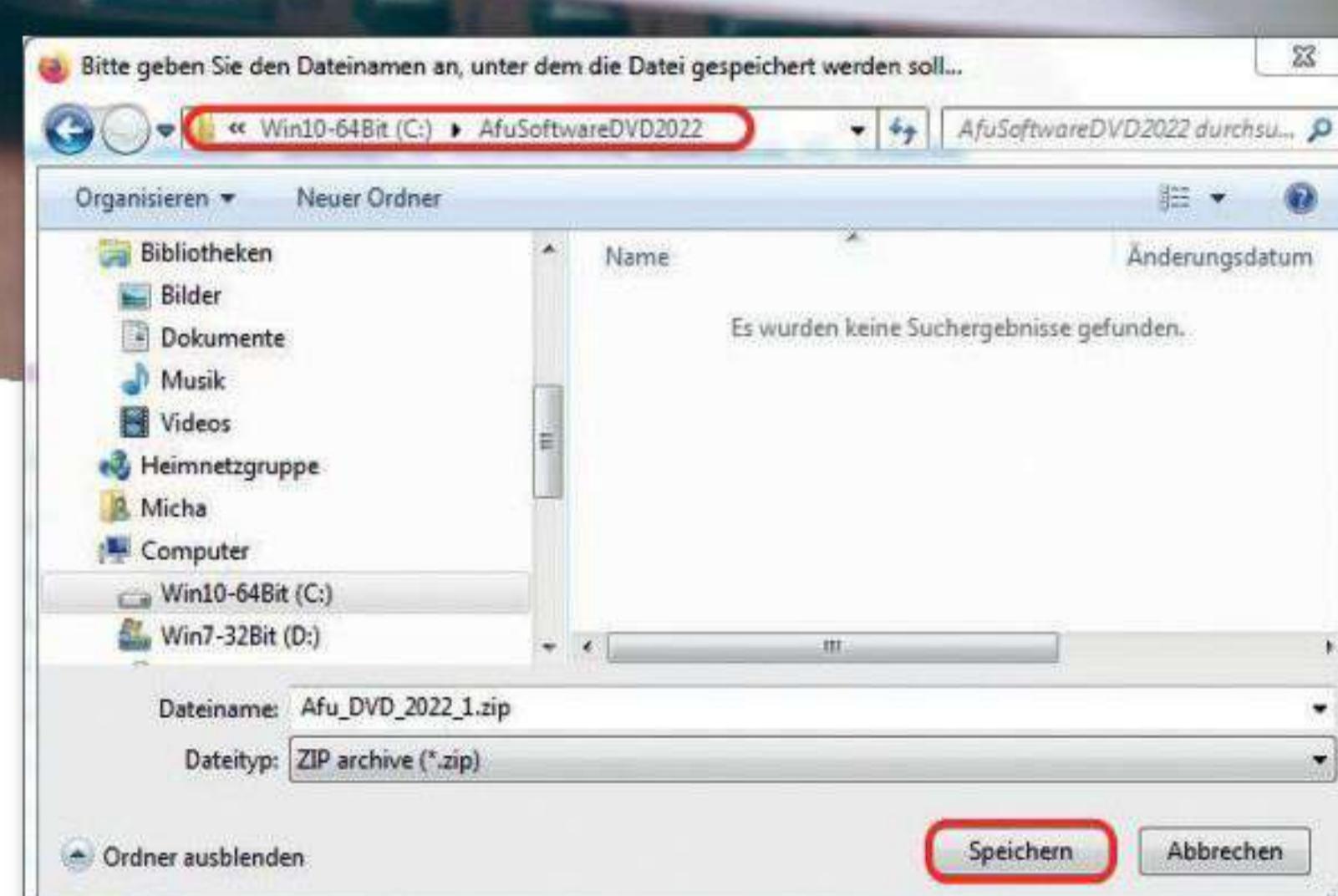
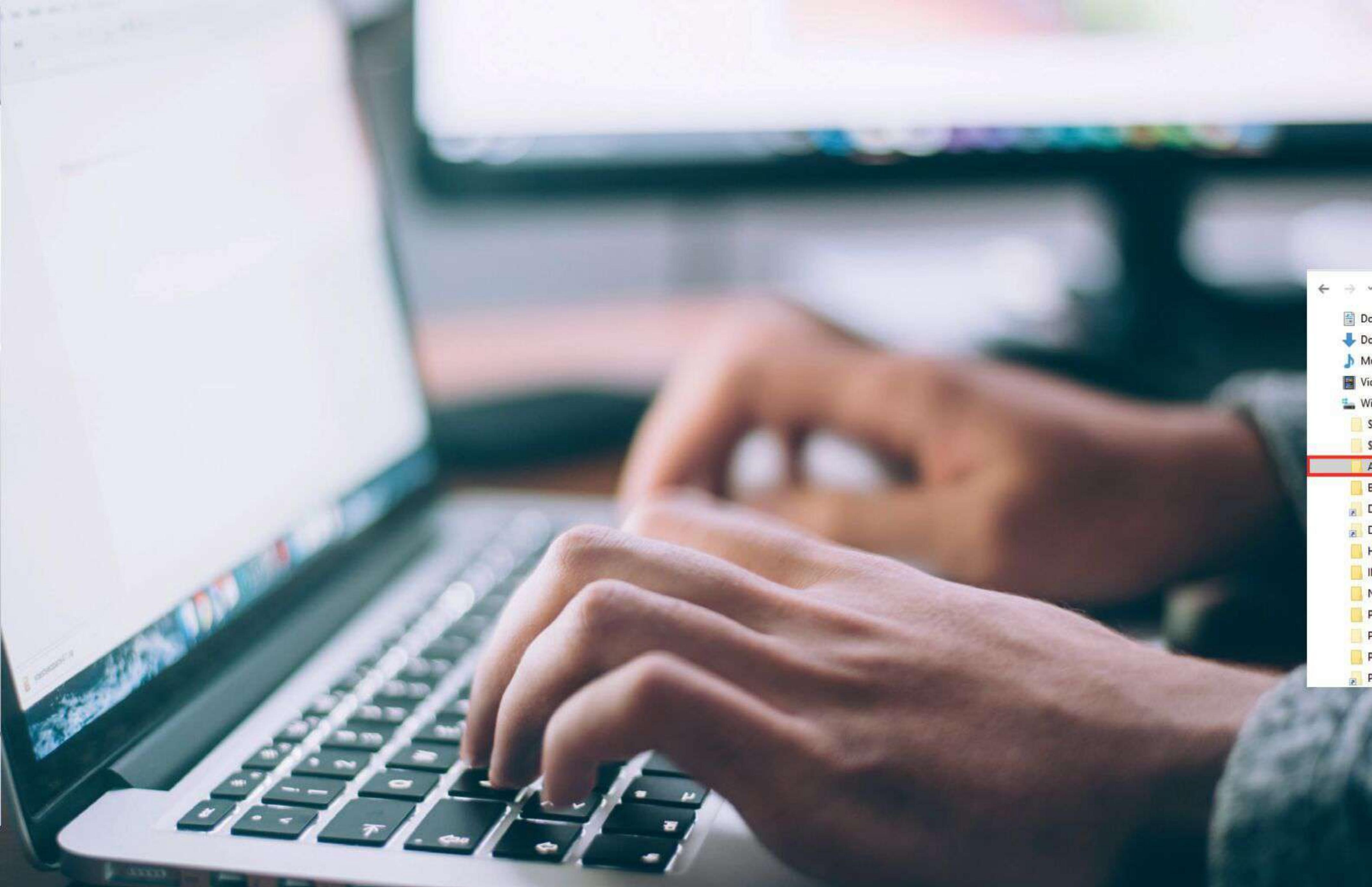
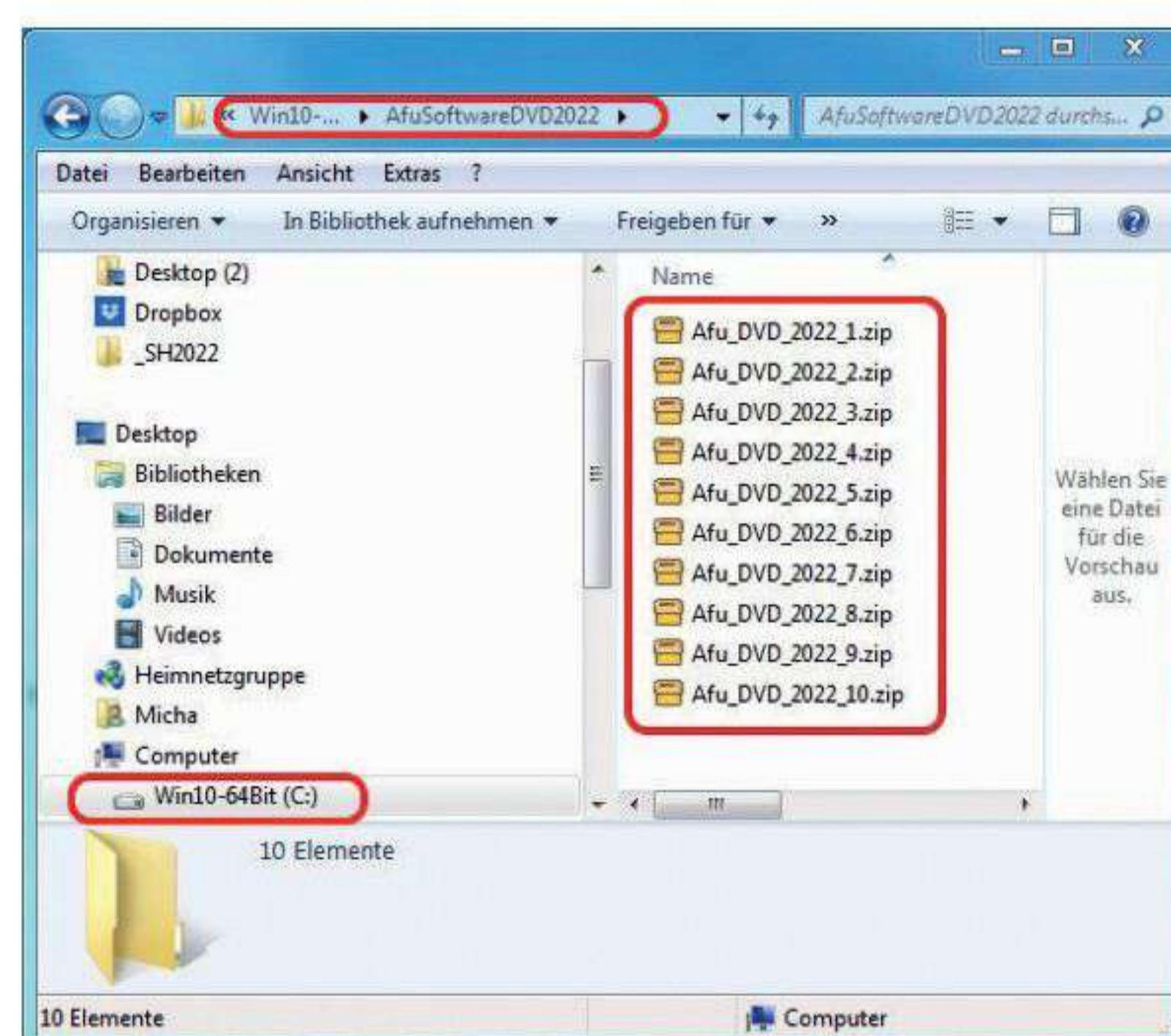


Bild 3: Geben Sie das Zielverzeichnis an. In dieses werden die Dateien abgelegt.



Schritt 3:

Laden Sie auf diese Weise alle Dateien auf Ihre Festplatte und legen Sie diese sämtlich in dasselbe Verzeichnis ab. Es ist wichtig, dass alle Dateien im selben Ordner versammelt werden! Bild 4 zeigt die geladenen Dateien aus dem Ordner *C:\AfuSoftwareDVD2022*. Hier sind es zehn Dateien, die Anzahl kann sich, wie gesagt, noch ändern.

Schritt 4:

Entpacken Sie nun die Zip-Dateien nacheinander in diesen Ordner. Mit den Mitteln von Windows geht das wie folgt: Markieren Sie eine Zip-Datei mit der Maus und betätigen Sie die rechte Maustaste (Rechtsklick auf die Zip-Datei). Bild 5 illustriert das - es öffnet sich das Kontextmenü. Wählen Sie die Option *Alle extrahieren*.

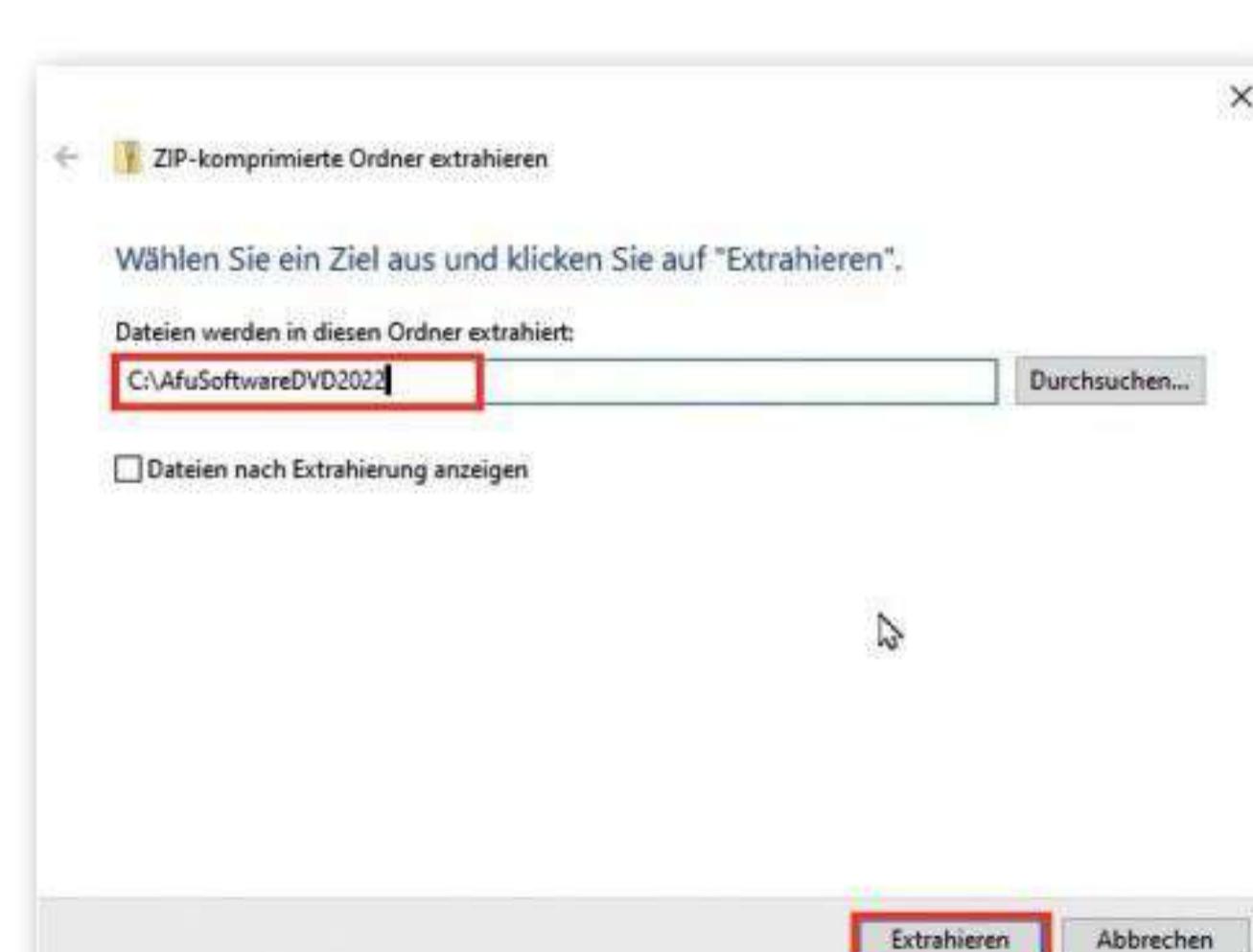


Bild 6: Legen Sie das Zielverzeichnis fest. Es ist wichtig, dass alle Dateien in dasselbe Verzeichnis der Festplatte abgelegt werden.

Verfahren Sie mit den weiteren Zip-Dateien wie zuvor beschrieben und geben Sie stets als Zielverzeichnis das in Bild 6 rot gerahmte Verzeichnis ein. Das garantiert, dass die in den Zip-Dateien vorgegebene Verzeichnisstruktur erhalten bleibt. Nachdem

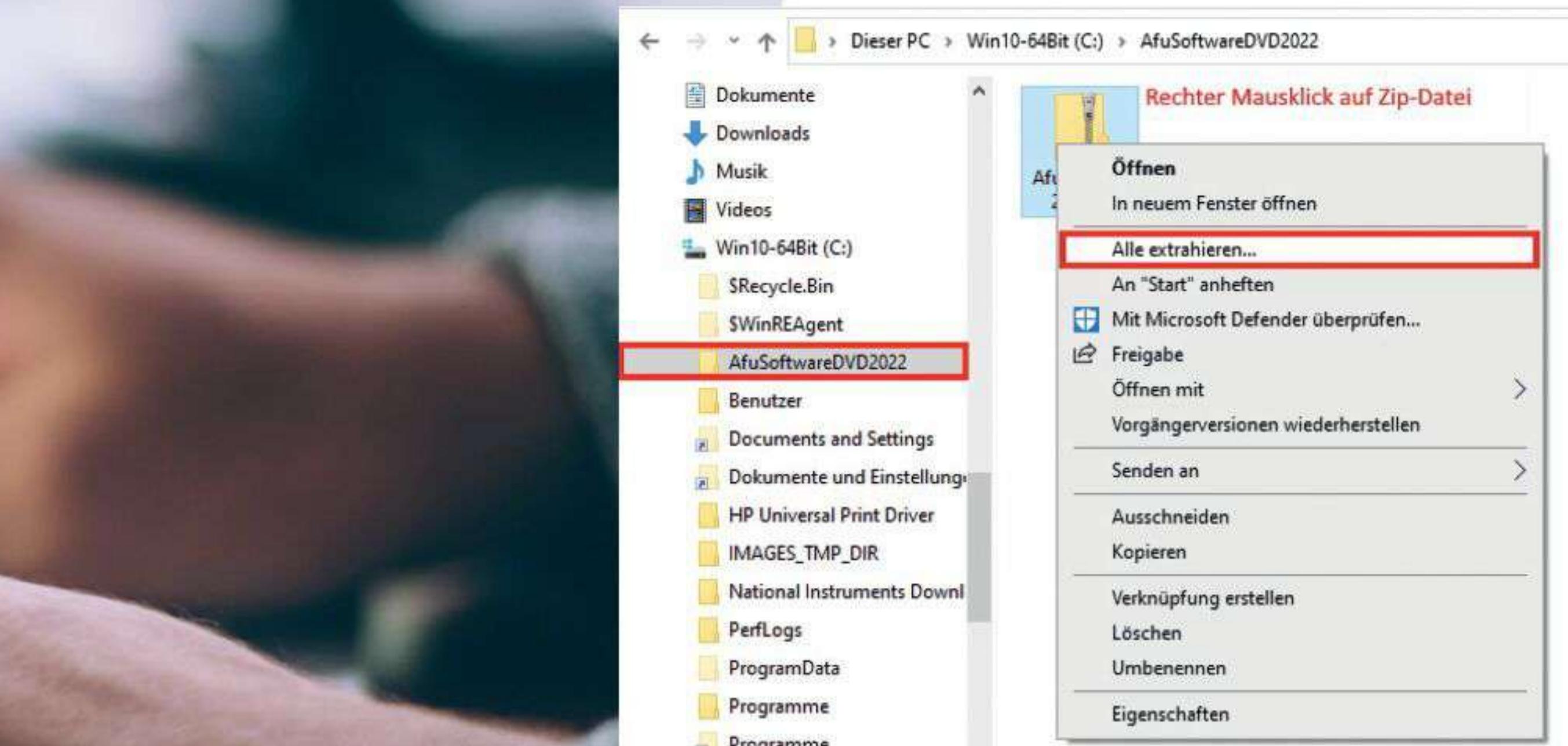


Bild 5: Nun geht es ans Auspacken der Zip-Dateien. Starten Sie mit einem rechten Mausklick auf eine der Dateien. Und wählen Sie *Alle extrahieren*.

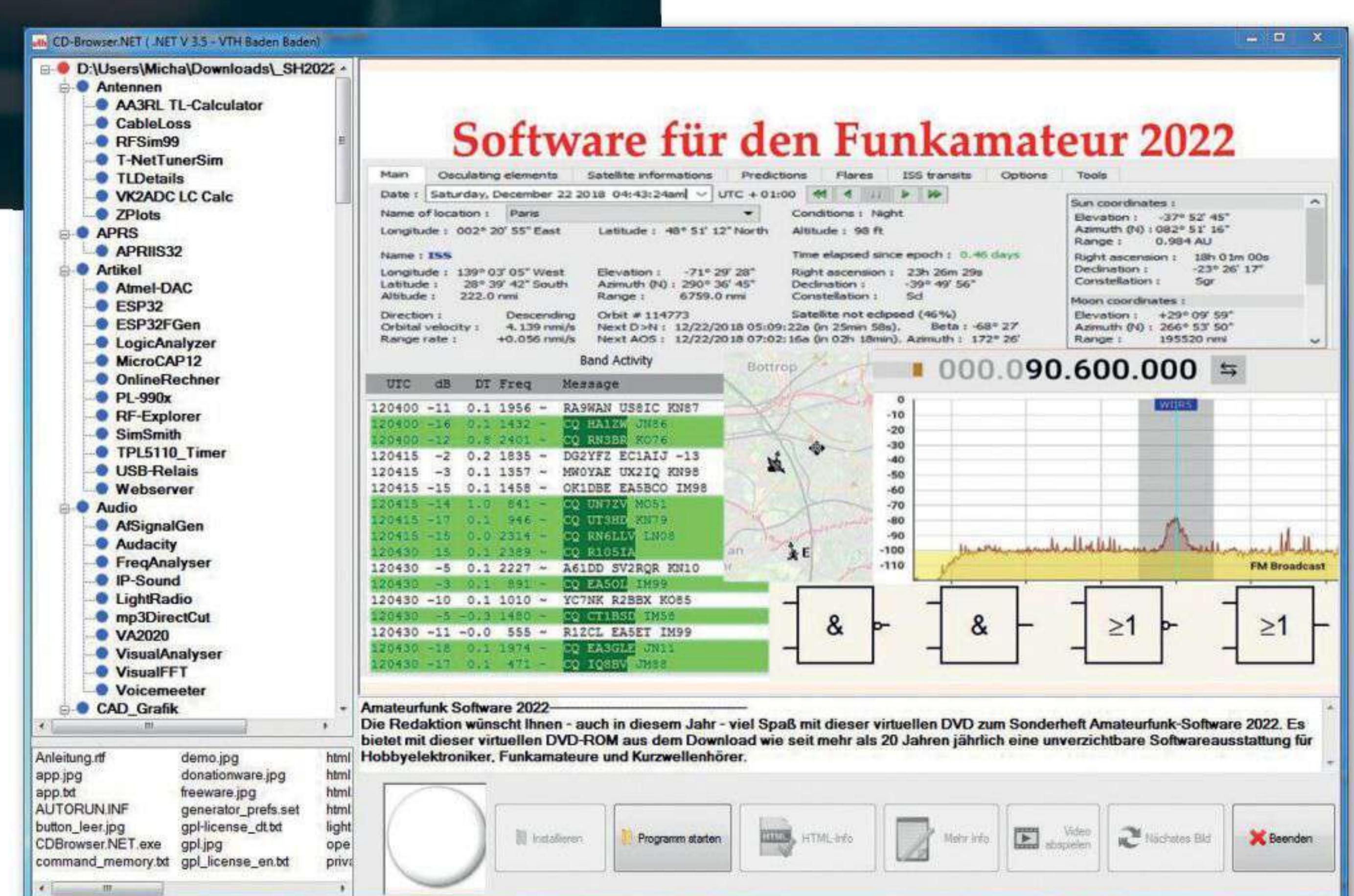


Bild 8: Nach dem Starten des CDBrowsers sieht es auf Ihrem PC-Monitor etwa so aus. Links sind die Rubriken und innerhalb der Rubriken die Programme zu sehen. Klicken Sie auf ein Programm, um zu jeder Software ein Bildschirmfoto und eine Beschreibung zu sehen.

Sie sämtliche Zip-Dateien auf diese Weise ausgepackt haben, finden Sie im Zielverzeichnis eine Reihe von Dateien und für jede Rubrik ein Verzeichnis sowie deren Unterverzeichnisse für jedes in einer Rubrik abgelegte Programm. Unter anderen ist dort auch der DVD-Browser mit der Bezeichnung *CDBrowser.Net.exe* abgelegt. Starten Sie diese Software (Bild 7), es öffnet sich der CD-Browser, mit dem Sie bequem durch alle Amateurfunkprogramme der virtuellen Heft-DVD

wandern und zu jedem Programm der DVD eine kurze Anleitung und eine Abbildung erhalten (Bild 8). Weiterhin können Sie aus dem CD-Browser heraus Software starten oder installieren und über die Schaltfläche *HTML-Info* interessante Webseiten aus der Rubrik *Links ins Internet* aufrufen. Auf diesen verlinkten Webseiten finden Sie es in der Regel weitere nützliche Programme zum Download. Fragen zum CD-Browser beantwortet Ihnen die Datei *Anleitung.rtf*.

Da es im letzten Jahr einige Anfragen zum Virenschutz gab, sei an dieser Stelle ein Hinweis erwähnt: Nicht jede Virenmeldung eines Antivirenprogramms ist real. Die hier zum Download angebotenen Dateien wurden zuvor mehrfach auf Virenfreiheit geprüft. Es ist jedoch nie falsch, bei Programmen besonders kritisch zu sein! Das gilt natürlich besonders dann, wenn sie aus dem Internet heruntergeladen werden. *mw*

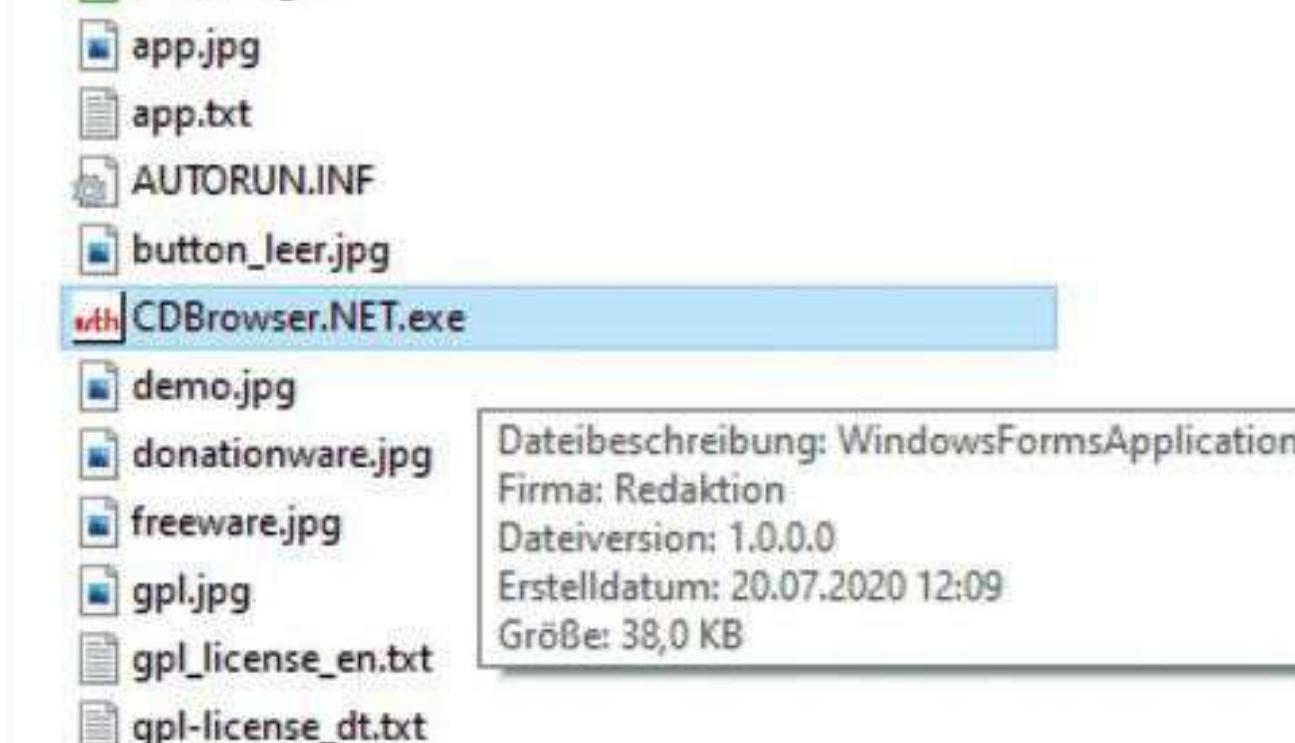


Bild 7: Sind alle Zip-Dateien ausgepackt, finden Sie diese EXE-Dateien im CDBrowser - im Verzeichnis.

Der RF-Explorer ist handlich und kompakt. Das Metallgehäuse macht einen stabilen Eindruck, die entfernbare Gummiarmierung schützt das Gerät vor unsanfter Behandlung. Die Höhe beträgt 120 mm, die Breite 75 mm und die Tiefe 30 mm. Ein Gerät, das bequem in eine geräumige Jackentasche passt. Auf der Oberseite befindet sich der SMA-Antennenanschluss (50 Ohm) und der Ein-Ausschalter als stabiler Schiebeschalter. Das Schwarzweiß-Display mit Hintergrundbeleuchtung bietet 128 x 64 Pixel, ist 50 mm breit und 25 mm hoch. Auch bei Sonnenlicht ist es gut ablesbar. Kontrast und Intensität der Hintergrundbeleuchtung lassen sich über ein Menü in mehreren Stufen justieren. Zur Bedienung dienen neun Tasten.

Geliefert wird es mit zwei Antennen, eine davon ist als Teleskopantenne für 144/430 MHz ausgelegt, die Gummiwendelantenne für den UHF-Bereich von 400 bis 960 MHz. Der Frequenzbereich dieser Gerätevariante reicht von 0,05 - 960 MHz, andere – kostspieligere – Geräte empfangen bis etwas über 6 GHz. Zum Lieferumfang gehört ein USB-Kabel zum Anschluss des Analysators an einen Computer und ein schickes Köfferchen zum Transport. Der RF-Explorer ist ein sehr empfindlicher Spektrumanalysator, der für die Erfassung von Signalen über die beiden Antennen entwickelt wurde. Für stärkere Signale als die absolute maximale Eingangsleistung von +30dBm (nur bei eingeschaltetem internem Dämpfungsglied!) ist stets darauf zu achten, dass der Eingang nicht überlastet wird. Der interne 30-dB-Abschwächer ist über ein Menü zuschaltbar. Stärkere Pegel müssen über ein externes Dämpfungsglied reduziert werden. Ein interner LNA (Low Noise Amplifier) mit 25 dB Verstärkung hilft extrem schwachen Signalen auf die Beine.

Das Display – klein, aber fein

Das Display ist durchaus informativ: Neben dem Spektrum, das die meiste Fläche einnimmt, passen unterhalb der X-Achse die Start-, Mitten- und Stopp-Frequenz, links einige Statusinformationen (z.B. die Batterieanzeige) und im Spektrum



Die Dichte elektronischer Geräte steigt insbesondere in urbanen Gebieten. Kleinere Grundstücke und eine Nachverdichtung der Bebauung bewirkt, dass Menschen und deren Geräte immer näher aneinanderrücken. Funkamateure klagen seit Jahren über eine steigende „Verschmutzung“ des Äthers. Ein mobiler Spektrumanalysator wie der RF-Explorer kann dabei helfen, einem störenden Gerät auf die Schliche zu kommen.

Bild 1: Der RF-Explorer, ein Spektrumanalysator bis 960 MHz.

Spektren entdecken mit dem RF-Explorer WSUB1G-Plus

für das stärkste Signal die Frequenz und der Wert der Amplitude auf diese begrenzte Zahl Pixel. Die Anzeige des Spektrums ist zwischen dBV und dBm umschaltbar. Bild 2 zeigt eine Abbildung mit vierfacher Pixelzahl. Als Anzeigemodi des Spektrums stehen zur Auswahl: Peak Max und Hold, Normal, Overwrite und der Averaging Modus. *Peak* zeigt stets das aktuelle Spektrum und darüber gemessene Maximalwerte an, *Max* zeigt nur die Kurve aus Maximalwerten des Spektrums, Im Modus *Normal* zeigt es das Spektrum ohne weitere Veränderungen. Im Modus *Overwrite* schreibt die Firmware neue Werte in die Grafik, ohne alte Werte zu löschen. Mit der Zeit ergibt sich ein breites Band des Graphen, die von minimaler Amplitude bis zur Maximalen reicht. Im *Average*-Modus wird das Spektrum aus dem

Durchschnitt mehrerer Werte (Iterationen) gemittelt und das Resultat zur Anzeige gebracht. In einem Menü ist die Zahl der Iterationen einstellbar.

Lassen die Einstellungen zur Darstellung des Spektrums kaum Wünsche offen, ist der Parameter

Marker aus dem Konfigurations-Menü eine wichtige Hilfe: Ist er auf *Peak* eingestellt, verhält sich die Anzeige wie oben beschrieben: Die Firmware sucht aus dem Spektrum das stärkste Signal (mit der höchsten Amplitude) und zeigt dazu Frequenz

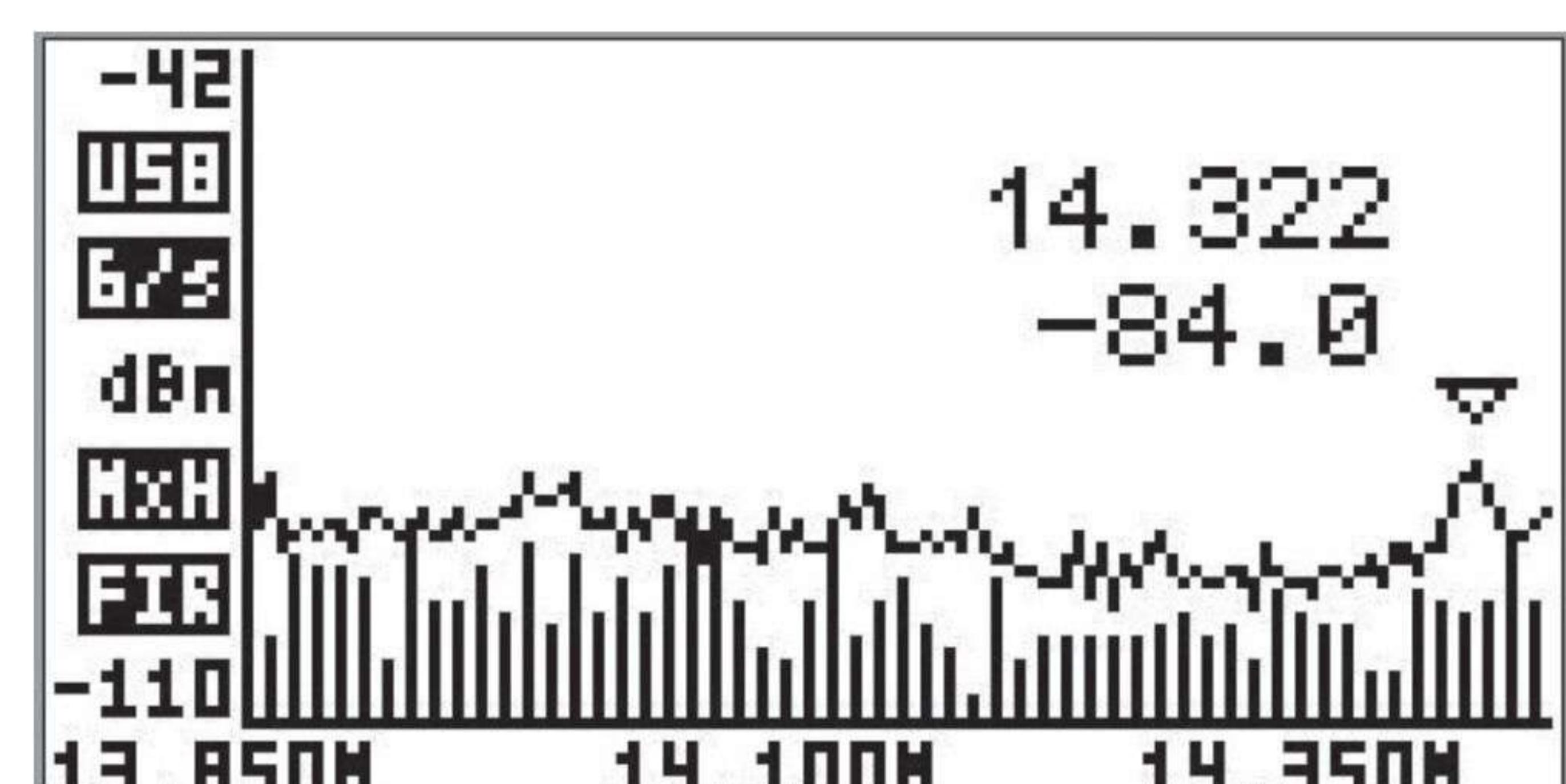


Bild 2: Diese Abbildung des LCD wurde mit der PC-Software über die Funktion *Remote Screen* erstellt und gibt den Inhalt der Anzeige komplett wieder.

und Amplitudenwert an. Doch nicht immer ist das stärkste Signal das interessante, sondern ein anderes auf einer anderen Frequenz. Um dessen Amplitude zu messen, stellt man den Marker im Menü auf *Manual* um, mit den Pfeiltasten rechts/links verschiebt man den Marker auf der X-Achse frei zur gewünschten Frequenz. Zu dieser werden auch hier Frequenz und Amplitude im Spektrum eingeblendet. Damit wird das kleine Gerät zum wertvollen Messmittel, da Frequenz und Amplitude für jede Frequenz exakt ermittelt werden kann – auch ohne PC-Anbindung.

Einige weitere Parameter lt. Herstellerangabe:

- Auflösung der Amplitude: 0,5dBm
- Dynamikbereich: -125dBm bis 10dBm
- Durchschnittlicher Rauschpegel (typisch LNA): -125dBm
- Frequenzstabilität und Genauigkeit (typisch): +/-10ppm
- Amplitudenstabilität und -genauigkeit (typisch): +/-2dBm
- Auflösungsbandbreite (RBW): automatisch 2,6Khz bis 600Khz

Der auf der Unterseite angeordnete USB-Anschluss dient – wie erwähnt – zur Verbindung mit einem PC, aber auch zum Laden des 1800-mA/h-Lithium-Akkus. Der reicht für mehr als 20 Stunden Dauerbetrieb aus.

Tasten und Presets

Der RF-Explorer wurde als autarkes Gerät für den mobilen Einsatz konzipiert. Die Bedienung ist weitgehend intuitiv. Das Stöbern in der englischen Anleitung gibt die nötigen Tipps: Über neun Tasten kommuniziert der Benutzer mit dem Gerät in über acht Menüs. Mit der Taste *Menü* gelangt man von einem Menü zum anderen. Mit den Pfeiltasten oben/unten wählt man einen Menüeintrag aus und mit *Enter* gelangt man in den Editiermodus zum Ändern eines Parameters und ebenso mit *Enter* wieder zurück. Im Editiermodus verändert man den Parameter – jede Ziffer separat – mit den Pfeiltasten oben und unten. Ein Menü verlässt man schnell über die Taste *Return*. Über die Taste *Help* liest man erwartungsgemäß einen

englischen Hilfetext, abhängig davon, in welchem Menü man *Help* betätigt hat. Hat man die Bedienung zwei Mal geübt, bedient sich das Gerät fast von selbst.

Weil es lästig ist, für eine Messung wiederholt Start- und Stopffrequenz oder alternativ Mittelfrequenz und Frequenzbreite (*Span*) anzugeben, bietet der Analysator das Speichern von zu messenden Frequenzbereichen zum Ablegen in sogenannten Preset-Speichern. Sie bieten sich für wiederholt genutzte Frequenzbereiche und deren Einstellungen geradezu an. Ein Preset ist aus der aktuellen Einstellung schnell gespeichert, lediglich das Zusammensuchen der Buchstaben zur Beschreibung des Presets dauert am Gerät eine Weile, da man für jeden Buchstaben mit den Pfeiltasten den Buchstaben suchen muss. Das Laden eines einmal abgelegten Presets geht dafür umso schneller und flugs ist der kleine Analysator wieder bei der Arbeit. Die Betätigung der Taste *SET* führt zum Preset-Menü und dort direkt auf den Eintrag *LOAD* – minimale Bedienungsschritte zum Laden eines Presets. Es sind insgesamt einhundert Presets möglich, mehr, als man sich im Kopf merken kann.

Die Software

Mit einer Steuertoolsoftware erhöht sich die Auflösung des Spektrums und auch das Verwalten der Presets

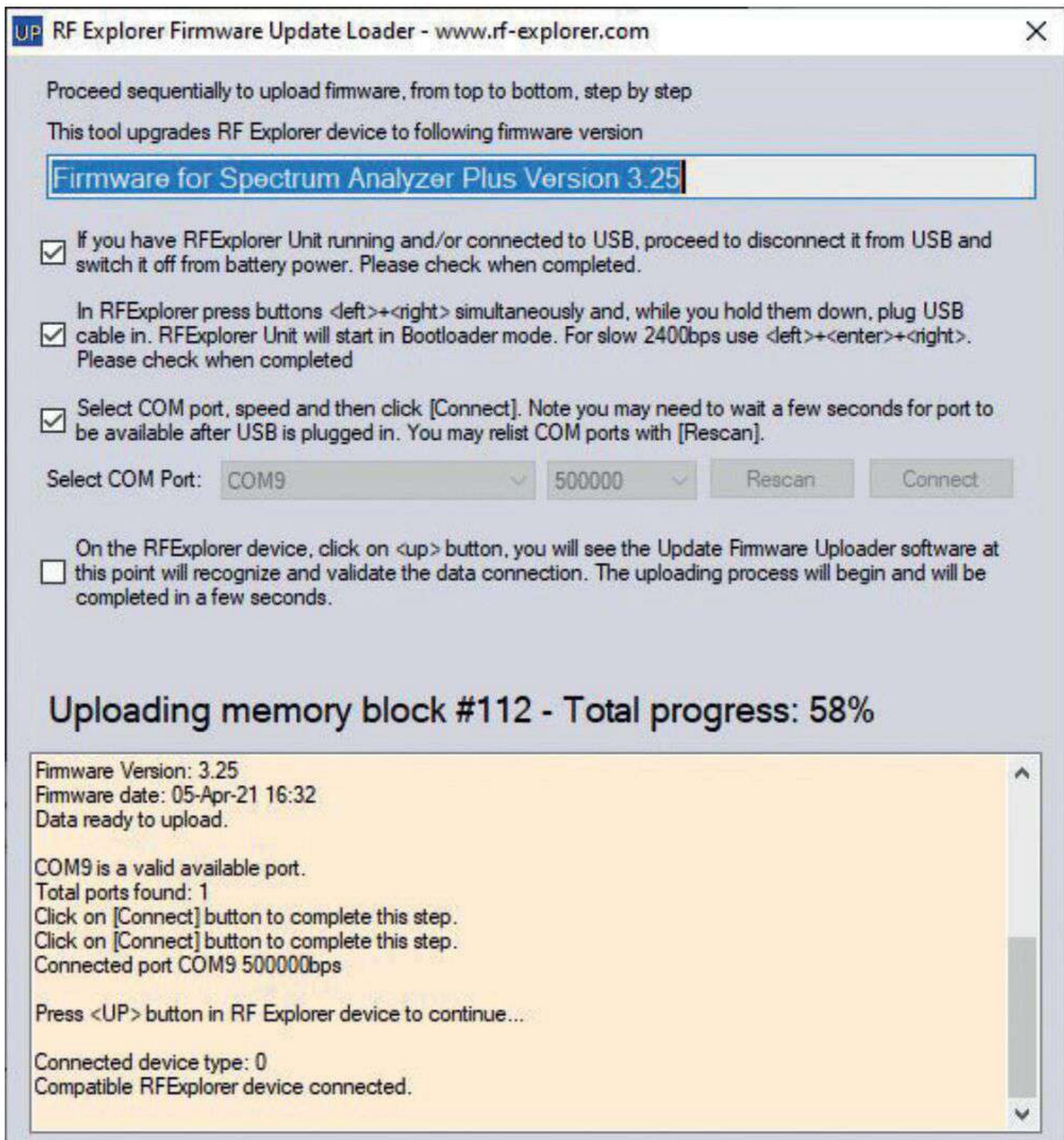


Bild 4: Mit 500.000 Baud wird das Update der Firmware aufgespielt. Es kamen neue Funktionen hinzu.

gestaltet sich dank PC-Tastatur und Maus wesentlich einfacher. Über eine PC-Software, die es für Windows, Linux und MacOS als Open-Source-Software gibt, lässt sich die begrenzte Auflösung des Displays erweitern und die Bedienung erleichtern. Eine API bietet Programmierern die Möglichkeit, eigene Software zu schreiben und den RF-Explorer zu steuern wie auch Daten zu empfangen. Das Pro-

gramm gilt für alle Gerätevarianten des RF-Explorers.

Die Software lädt man von der Geräte-Homepage [1] in der jeweiligen Version für das benutzte Betriebssystem (hier Windows 10). Nach der Installation erfährt man beim ersten Start, dass dieses Open-Source-Programm erst seinen Dienst nach einer Registrierung antritt. Es war schon tiefen Nacht, die Regis-

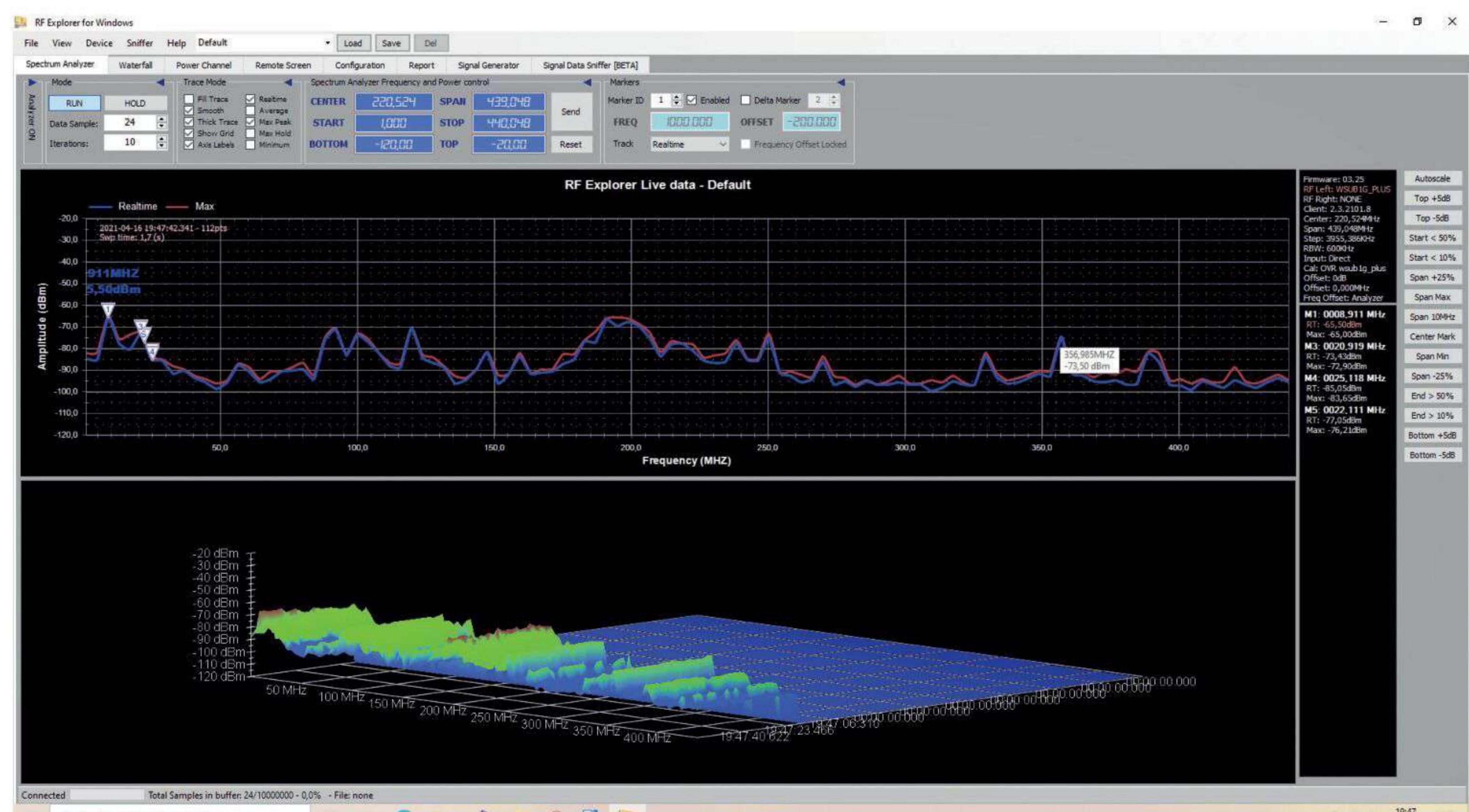


Bild 3: Blick auf die Windows-Software im dark mode. Oben ist das Spektrum, unten die Wasserfallanzeige zu sehen. Letztere gibt es auch in der gewohnten 2D-Form.



Bild 5: Die Feldstärkeanzeige für Amateurfunkbänder gibt es ab der Firmware-Version 3.25 als kostenpflichtige Zusatzfunktion.

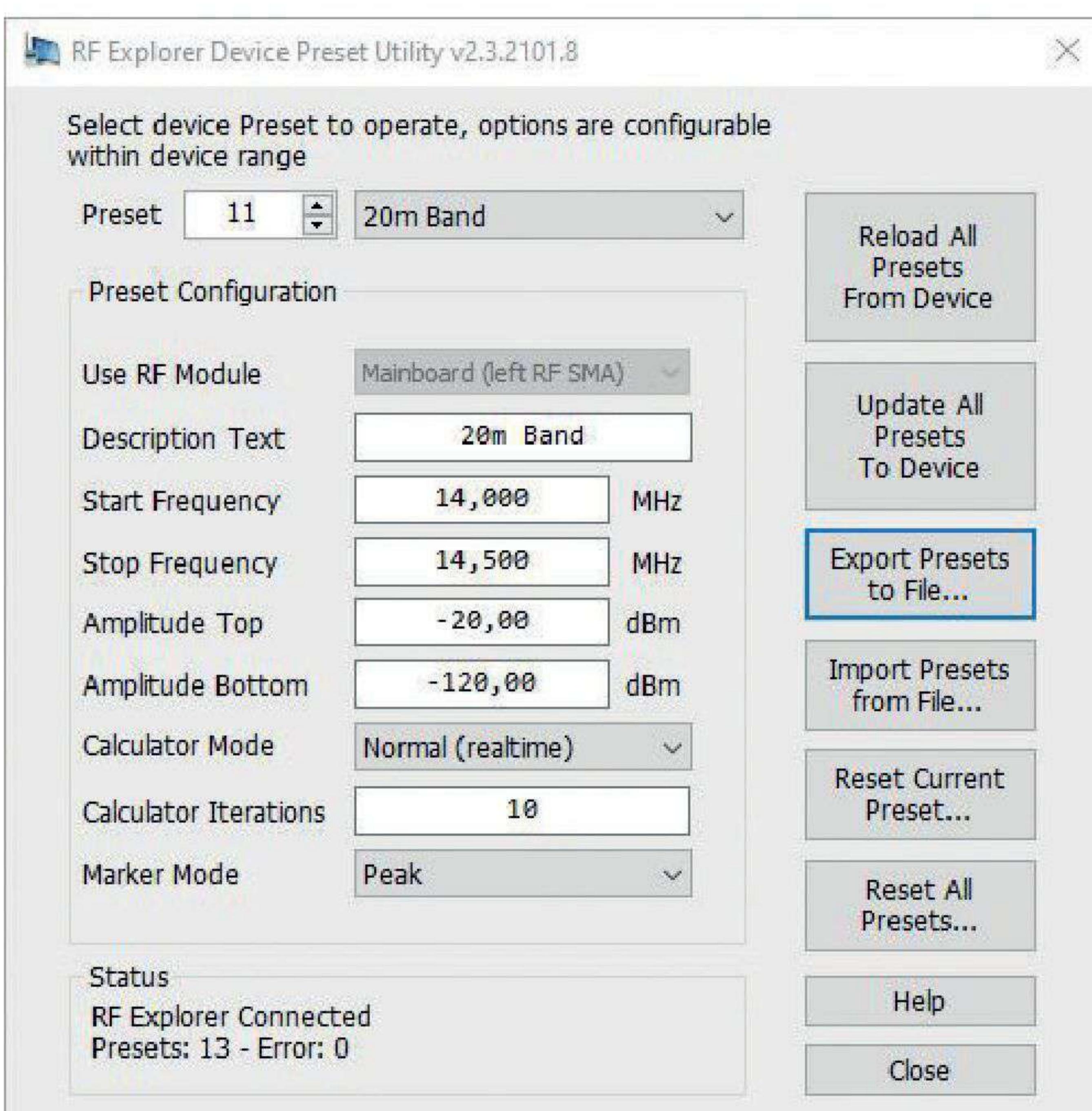


Bild 6: Der Preset-Manager erleichtert das Anlegen und Editieren von Presets enorm.

rierung mit Nennung einer E-Mail-Adresse funktionierte einwandfrei. Die E-Mail des Registrierungsservers lies leider eine halbe Stunde auf sich warten, es dauert halt, bis die

Mail über einige Zwischenschritte im eigenen elektronischen Postfach eintrifft. Das Eintragen einer langen Buchstabenreihe über die Zwischenablage schaltete das Programm

frei. Ist der Analysator am PC angeschlossen, überprüft die Software die Aktualität der Firmware und fordert dazu auf, die neueste Version auf den Analysator zu übertragen. Dazu dient ein Extra-Programm, das mit der Windows-Software des RF-Explorers installiert wurde.

Firmware ja erst nach dem Kauf auf das Gerät gespielt wurde und kaum einen Monat alt war. Der Vertrieb konnte davon also keine Kenntnis haben bzw. nicht so schnell darauf reagieren.

Auf eine interessante Erweiterung wird allerdings bereits vor dem Kauf auf der Geräte-Homepage [1] hingewiesen: Drei zusätzliche Platinen, von denen eine in das Gerät eingebaut werden kann, ist mit einem zweiten Antennenanschluss ausgestattet und erweitert den Frequenzbereich des Spektrumanalysators lückenlos bis wahlweise 4 GHz oder 6.1 GHz. Lediglich die Zusatzplatine für 2,5 GHz erweitert lediglich um den Bereich von 2350 bis 2550 MHz. Bei dieser Zusatzplatine verbleibt eine Frequenzlücke zwischen 960 und 2350 MHz.

Doch zurück zur Windows-Software: Wie erwähnt, bietet sie eine grafisch ansprechende Bedienoberfläche. Ein Manager verwaltet komfortabel alle Presets und ermöglicht es, jedem Preset eine aussagekräftige Beschreibung auf den Weg zu geben, neue Presets zu definieren, diese an das Gerät zu übertragen, zu lesen und als XML-Datei auf dem PC zu sichern und zu laden.

Der RF-Explorer für Windows zeigt das Spektrum detailliert im Hires-Modus mit 512 Punkten auf der X-Achse. An zu beobachtenden Frequenzen setzt der Anwender einen oder mehrere Marker. Zwischen zwei Markern lässt sich die Differenzfrequenz berechnen. Fährt der

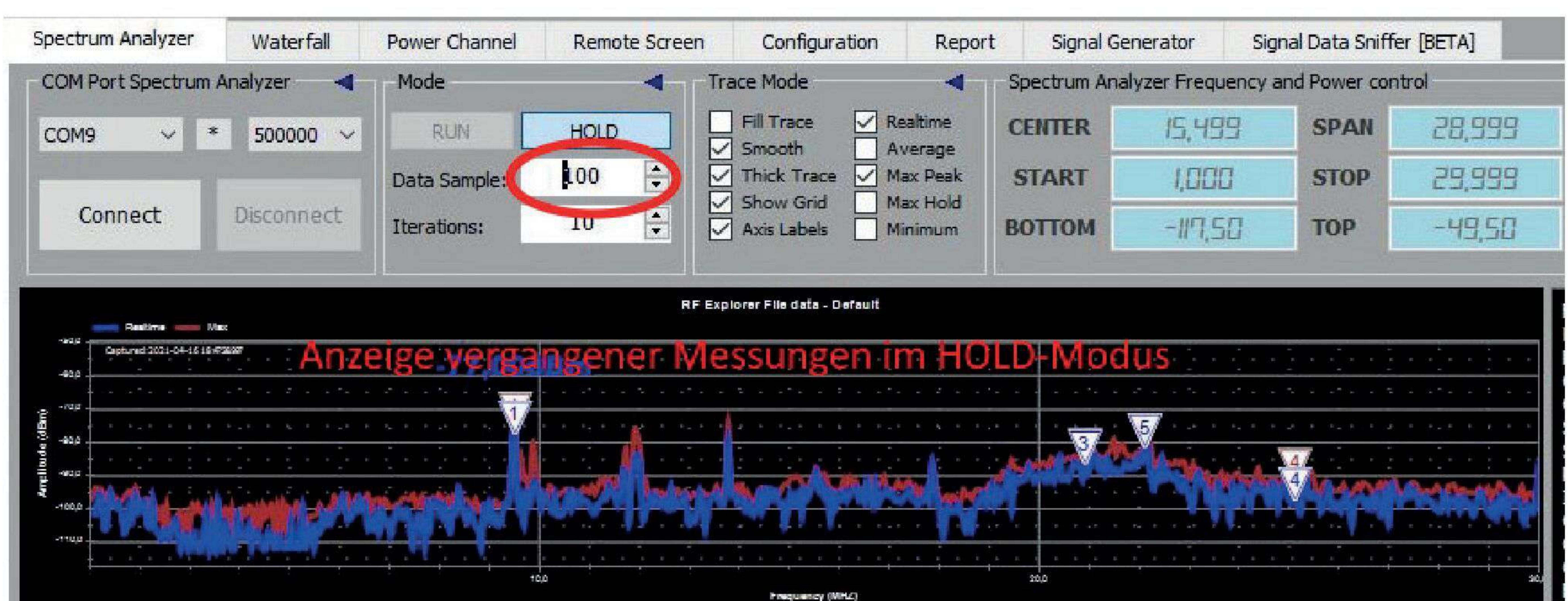


Bild 7: Im Hold-Modus „blättert“ der Anwender durch zuvor gemessene Samples. Dazu verändert man den rot markierten Wert Data-Sample. Es wurden fünf Marker gesetzt.

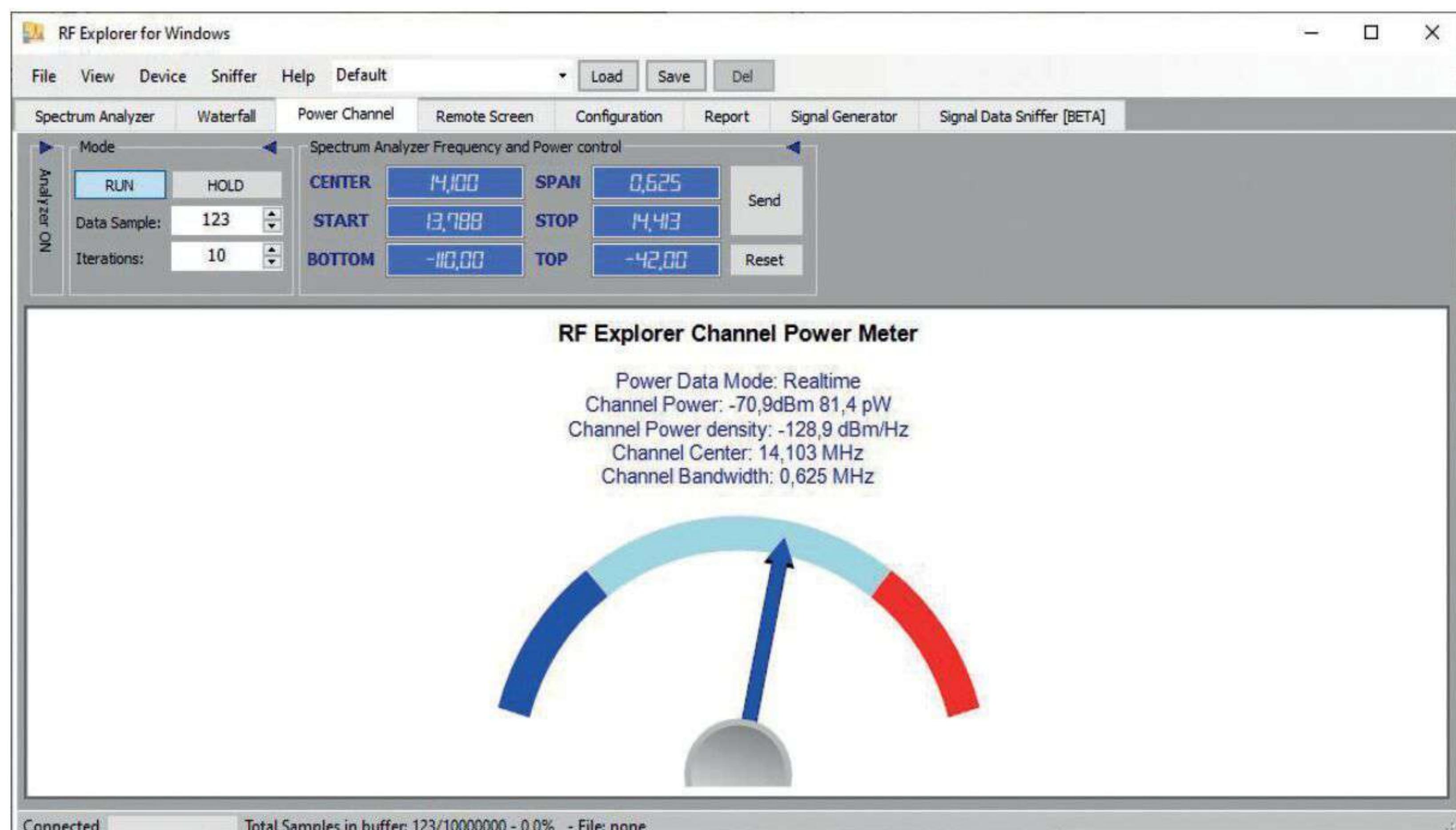


Bild 8: Leistungsanzeige der Windows-Software.

Benutzer mit der Maus entlang des Graphen, klebt an der Mausposition das Tooltip mit Angabe zu Frequenz und Amplitude. Voraussetzung dazu ist, dass die Option *Show Point Values* aktiviert ist, ein Standardwert der Konfiguration.

Schaltet man vom *RUN* in den *HOLD*-Modus um, stoppt die Messung. Die im Spektrum zu sehenden Daten (engl.: sweeps) lassen sich in einer RFE-Datei speichern und später wieder aufrufen und im *HOLD*-Modus des Gerätes Sample für Sample in aller Gemütlichkeit durchwandern und mehrmals erneut betrachten. Sehr praktisch ist das, um kurze Aussenungen z. B. eines Temperatursensors nachträglich auszumessen. Im *HOLD*-Modus blättert man auf diese Weise ein wenig zurück in die „Vergangenheit“.

Unterhalb des Spektrums ist die Anzeige des Wasserfalls angeordnet (Bild 3), die sich über ein Kontextmenü von 2D auf 3D und darin auf verschiedene Perspektiven umschalten lässt. Rechts neben Spektrum und Wasserfall sind eine Reihe von Buttons angeordnet, die sich direkt auf die Darstellung des Spektrums auswirken: Eine automatische Justage bewirkt *Autoscale*, andere Schaltflächen wirken sich auf die Skalierung der X- und Y-Achse aus. Beispielsweise ist es möglich, mit einem Mausklick die Bandbreite der Betrachtung auf 10 MHz einzustellen.

Daten lassen sich – wie beschrieben – als im eigenen RFE-Format speichern, aber auch als CSV, um sie z. B. In Excel oder ähnlichen Pro-

grammen weiter zu bearbeiten und grafisch zu verwerten. Abbildungen des Bildschirms – des Spektrums oder des Wasserfalls – kann man leicht als Grafik speichern oder in die Windows-Zwischenablage kopieren. Eine Druckfunktion ist ebenso vorhanden.

Beispiel einer Messung

Es soll der Rauschflur außerhalb der Ortschaft im Wald und im Wohnviertel in der Wohnung gemessen werden. Dazu wurden mit dem RF-Explorer jeweils freie Frequenzen gemessen und die Werte in dBm notiert. Zum Vergleich habe ich zusätzlich die Teleskopantenne über das Gehäuse des Internet-Routers gehalten. Hier die gemessenen Werte:

Insgesamt sind die Ergebnisse zufriedenstellend, lediglich beim 70cm-ISM-Band zeigt sich eine Differenz zwischen Wald und Wohnung von 4 dBm und zwischen Wohnung und Router von weiteren 10 dBm. Das sind immerhin 10 mW.

Statt des RF-Explorers – das mag mancher Leser denken – kann man alternativ mit einem SDR-Empfänger ähnliches erreichen. Das mag zwar richtig sein, solange dieser klein und handlich ist und die gewünschte Bandbreite aufweist. Wer sich nicht scheut, mit Laptop, z.B. AirSpy oder SDR-Play, Antenne und dem dazu benötigten Kabelsalat auf Störersuche oder Entdeckungsreise zu gehen, benötigt den RF-Explorer in der Tat nicht. Alle anderen, die einen handlichen, autarken Spektrumanalysator für die Jackentasche

vorziehen, sind mit diesem Gerätchen gut bedient.

International wird dieser RF-Explorer (ohne Erweiterungen) zur Zeit der Niederschrift dieses Beitrags für US\$ 169,00 bei [2] und in Deutschland für ca. € 210,00 bei [3] und [4] angeboten. Weitere Bilder zu diesem Beitrag finden Sie auf der virtuellen DVD im Ordner \Artikel\RF-Explorer.

Literatur/Verweise:

- [1] Homepage des Gerätes: <http://rf-explorer.com>
- [2] Vertrieb international: <https://www.seeedstudio.com/>
- [3] Vertrieb Deutschland: <https://www.wimo.com/de/>
- [4] Vertrieb Deutschland: <https://www.exp-tech.de/rf-explorer/>

Michael Wöste, DL1DMW

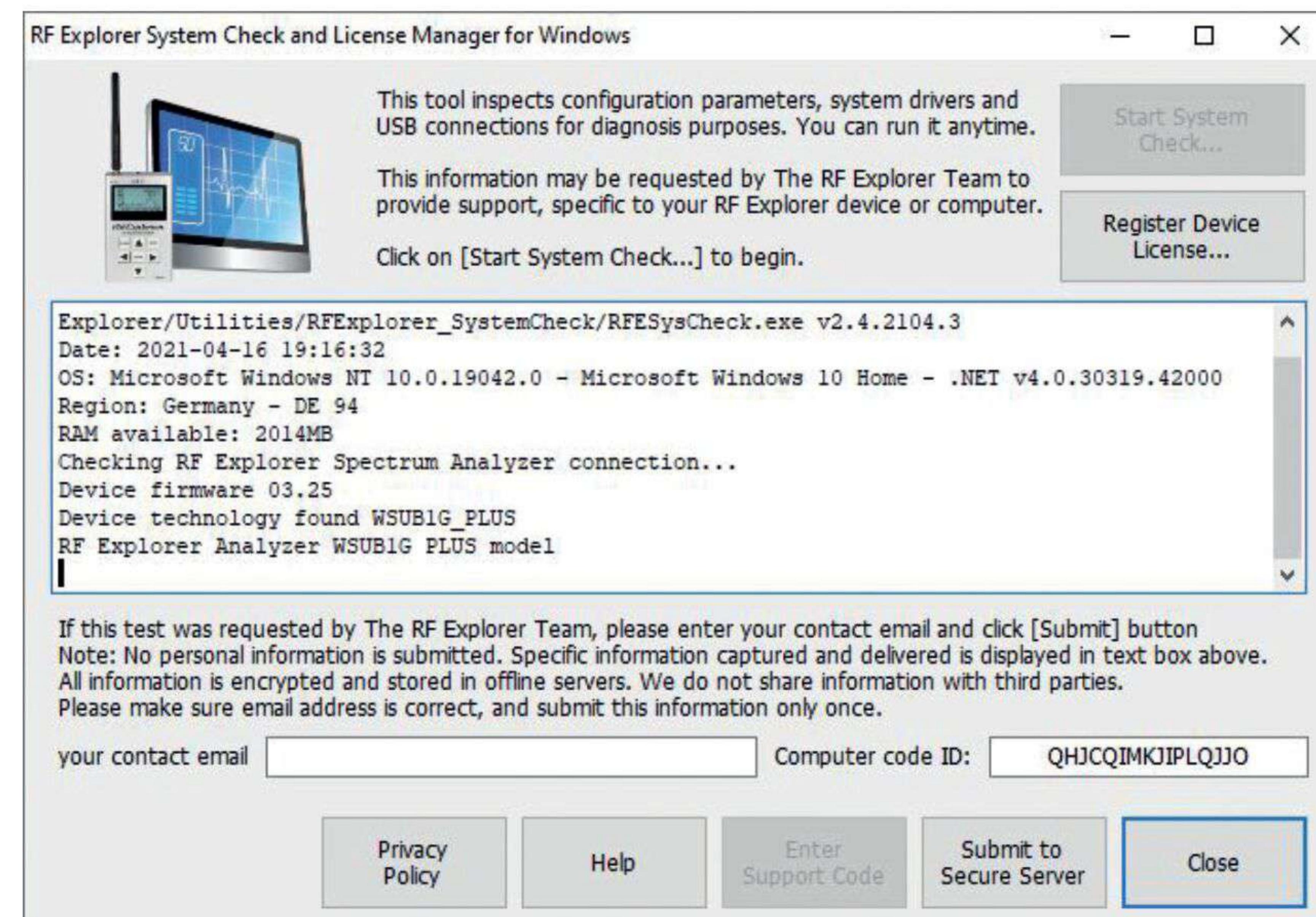
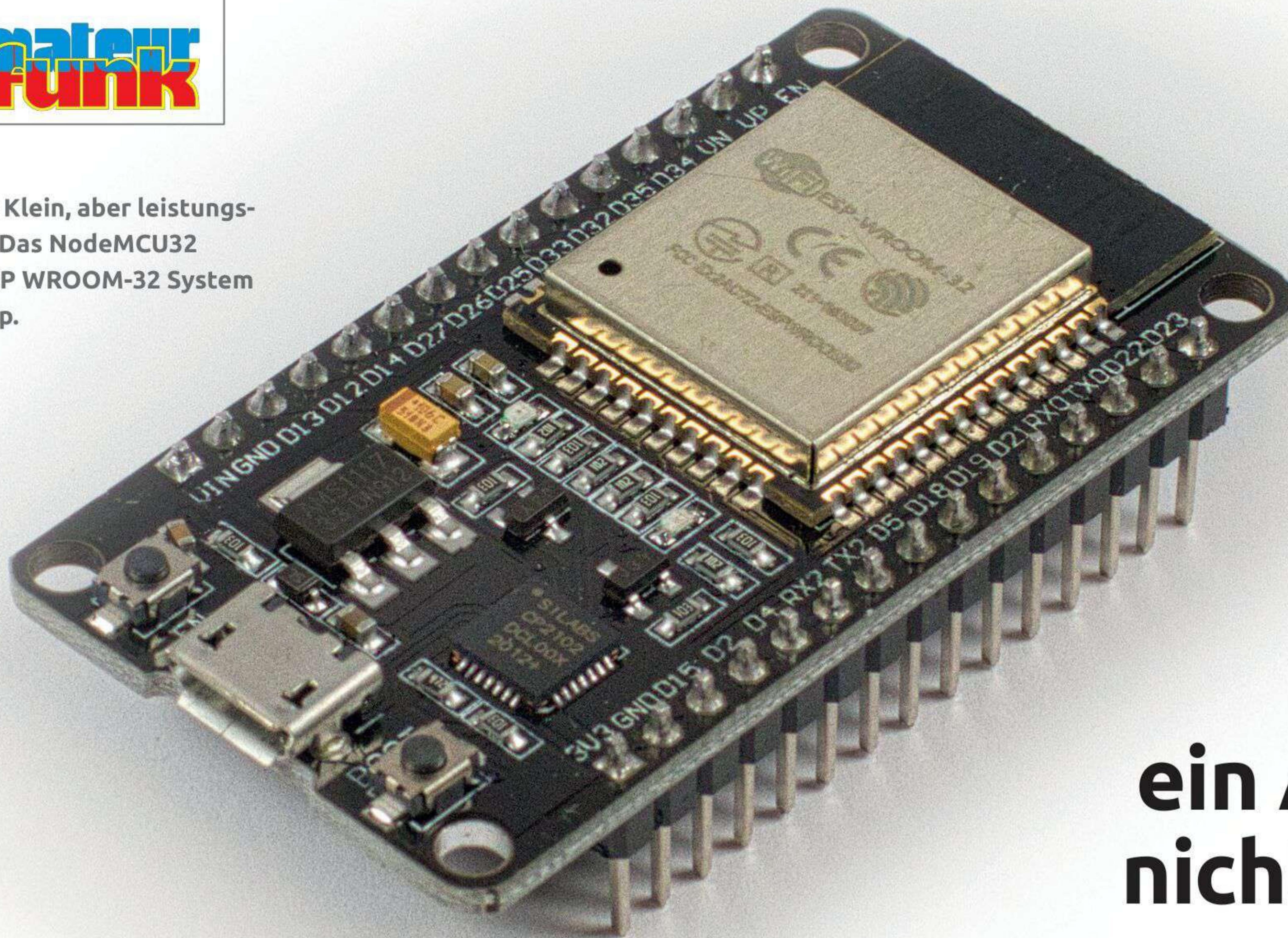


Bild 9: Ein System-Check hilft dem Support bei der Fehlerfindung.

Bild 1: Klein, aber leistungsstark: Das NodeMCU32 mit ESP WROOM-32 System on Chip.



Wenn
ein Arduino
nicht reicht:

NodeMCU mit 32-Bit-Controller

NodeMCU32 mit dem ESP32-WROOM-32 SOC (System on Chip) an Bord ist ein leistungsstarkes Modul, es zielt auf leistungshungrige Anwendungen wie Sensornetzwerke, Sprachkodierung, Musik-Streaming, MP3-Dekodierung und auf Anwendungen im Internet der Dinge (IoT - Internet of Things). Es integriert zwei CPU-Kerne, die CPU-Taktfrequenz ist einstellbar von 80 bis 240 MHz. Der stromsparende Co-Prozessor, der anstelle der CPU verwendet werden kann, dient dazu, Aufgaben zu erledigen, die nicht viel Rechenleistung benötigen. ESP32 integriert eine Vielzahl von Peripheriegeräten, wie den integrierten Hall-Sensor, eine Schnittstelle zu SD-Karten, Ethernet, bietet 4x SPI, 3x UART, 2x I²S und 2x I²C. Sogar eine CAN-Schnittstelle für Industrieanwendungen ist mit von der Partie. Das Modul weist 18 digitale GPIO-Pins auf, zwei 8-Bit-DACs, einen 12-Bit-ADCs mit Multiplexer, zusammen 18 ADC-Eingänge. Eine 16-Kanal-PWM steuert allerlei Geräte wie Motoren, LEDs u.v.m. Das Modul enthält 512 KB externes SRAM und 4 MB Programmspeicher. Die Betriebsspannung beträgt 3,3 Volt.

Das WiFi-Modul auf dem Chip ermöglicht eine Verbindung mit dem WLAN und über einen Router zum Internet. Das integrierte Bluetooth-Modul ermöglicht die Kommunikation mit anderen Bluetooth-Geräten wie Smartphones,

Arduino baut überwiegend auf 8-Bit-Prozessoren von Microchip, ehemals Atmel. Wer mehr Leistung benötigt, wählt einen 32-Bit-Mikrocontroller mit höherer Taktrate. Der NodeMCU32 mit ESP-WROOM-32-Chip von Espressif dient als Beispiel für einen leistungsstarken Prozessor mit reichlicher Ausstattung, die auch für den Funkamateuer interessant sein kann.

Kopfhörern u.a. ESP32 werkelt Strom sparend: Der Ruhestrom beträgt weniger als 5 µA, prädestiniert für batteriebetriebene und tragbare Geräte.

Der ESP32 verfügt intern über einen 8-MHz-Oszillator und einen RC Oszillator für externe Oszillatoren von 2 MHz - 60 MHz - für WiFi (WLAN) ist ein 40-MHz-Oszillator vorgeschrieben, der auch auf diesem Modul verwendet wird. Eine RTC (real time clock) kann man extern mit einem 32-kHz-Quarz bestücken. Weiterhin stehen vier 64-Bit-Timer für allgemeine Anwendungen zur Verfügung sowie mehrere Watchdog-Timer.

Der ESP32 verfügt über weitere, bisher hier nicht genannte Funktionen, doch schon die bisherige Aufzählung zeigt die ungewöhnlich

gute Ausstattung des Controllers. Da wundert es nicht, dass der Controller in der Programmierer-Community eine intensive Unterstützung erfährt und einen weiten Anwendungsbereich abdeckt.

Zur Programmierung und Fehlersuche gibt es vom Hersteller Espressif eine offizielle Entwicklungsumgebung unter der Bezeichnung *Espressif IoT Development Framework* (ESP-IDF). Dieses erweitert man in der Regel durch das „Einklinken“ von Komponenten, welche die Programmierung vereinfachen. Programmieren gelingt mit Programmiersprachen wie C, Python und anderen. Den deutlich leichteren und eher mühelos zu nennenden Zugang bietet hingegen die vielen Funkamateuren bekannte Arduino IDE.

Leichter Programmieren mit Arduino

In der Tat bietet die Arduino-Entwicklungsumgebung nicht nur Zugang zu den Arduino-Platinen, sondern unterstützt auch allerlei „fremde“ Prozessoren, darunter auch den ESP32. Dafür stellt Espressif die benötigten Routinen bereit. Zur Installation des ESP32 in die Arduino IDE geht man wie folgt in einigen wenigen Schritten vor:

Installieren Sie die Arduino IDE von [1], falls noch nicht geschehen.

Im Menü *Datei / Voreinstellungen* geben Sie in das Feld *Zusätzliche Boardverwalter-URLs* folgenden Text ein:

https://dl.espressif.com/dl/package_esp32_index.json

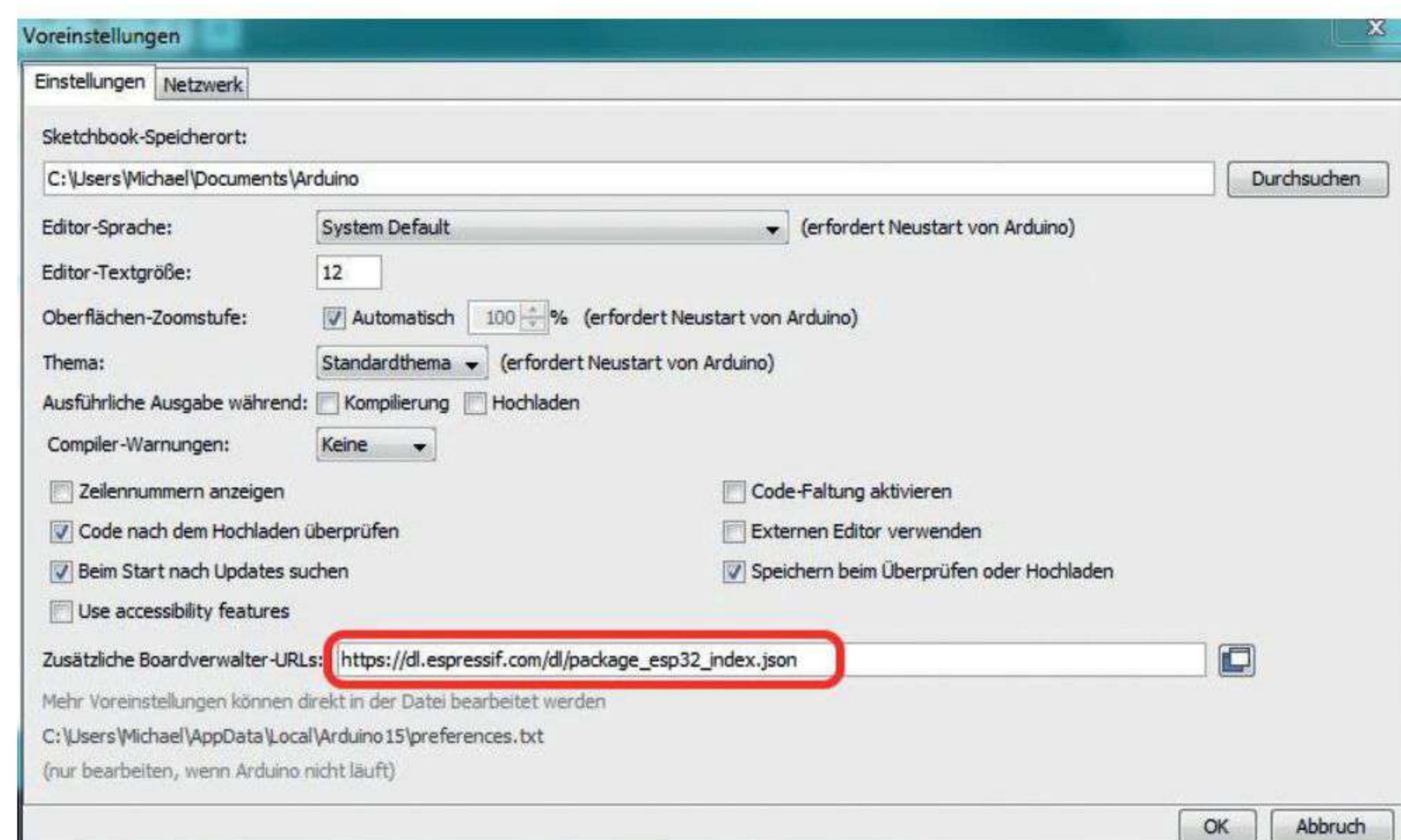


Bild 2: Zur Vorbereitung der Installation wird diese Zeile in den Dialog **Voreinstellungen** eingetragen. Das garantiert, dass die **ESP32-Platinen** im **Boardmanager** sichtbar werden.

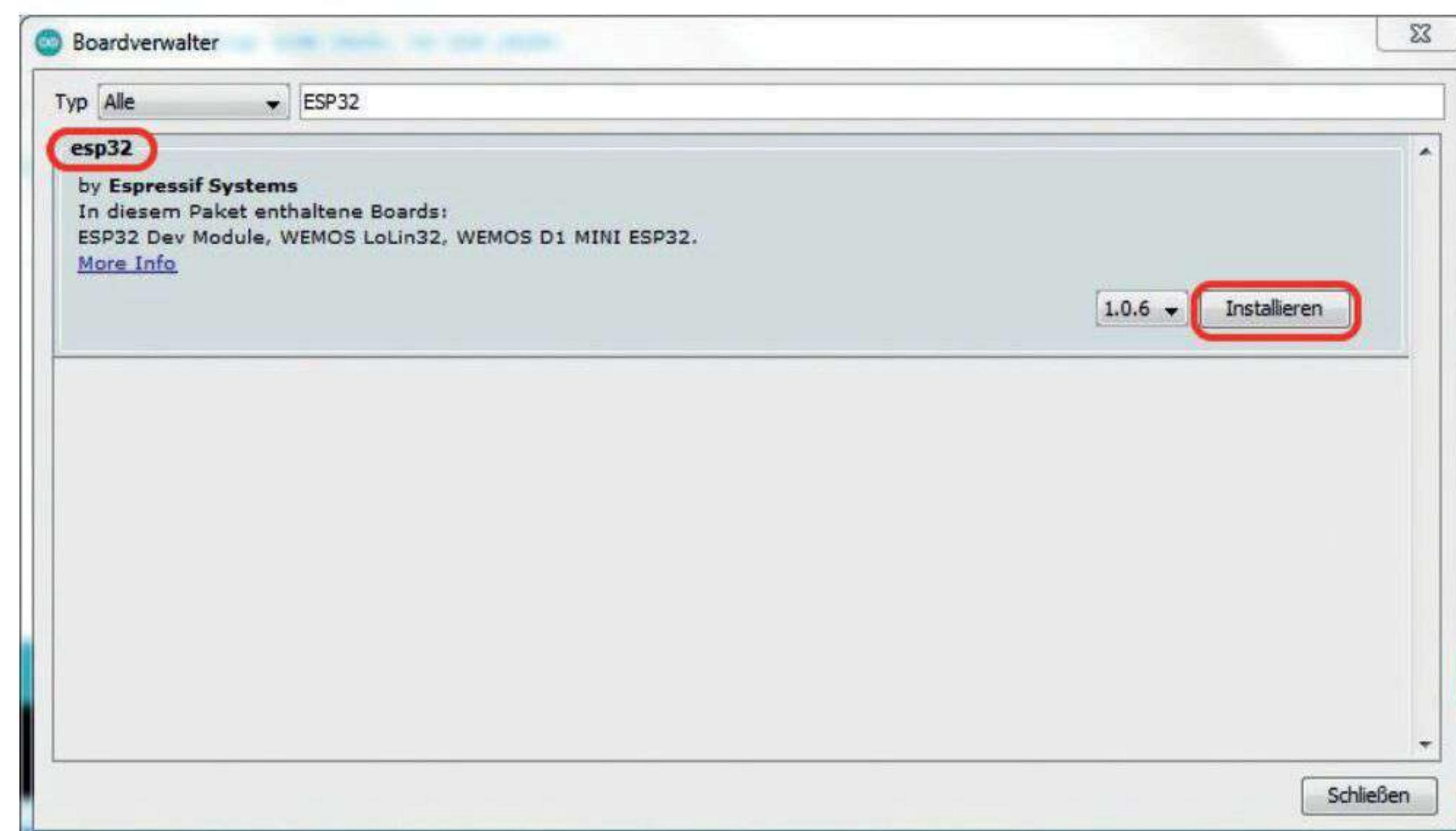


Bild 3: Im **Boardmanager** filtern man nach **ESP32** und installiert das Paket.

Rufen Sie nun den *Boardverwalter* aus dem Menü *Werkzeuge* auf. Sie finden ihn, wenn Sie das Board ändern möchten, dort in der aufpoppenden Liste aller Platinen ganz oben.

Sie das hier verwendete Modul, wählen Sie *ESP32 Dev Module* aus der Liste aus. Schließen Sie nun das Modul über ein passendes USB-Kabel an den PC an. Windows 7 bis 10 installiert den Gerätetreiber und weist dem ESP32-Modul einen virtuellen COM-Port zu.

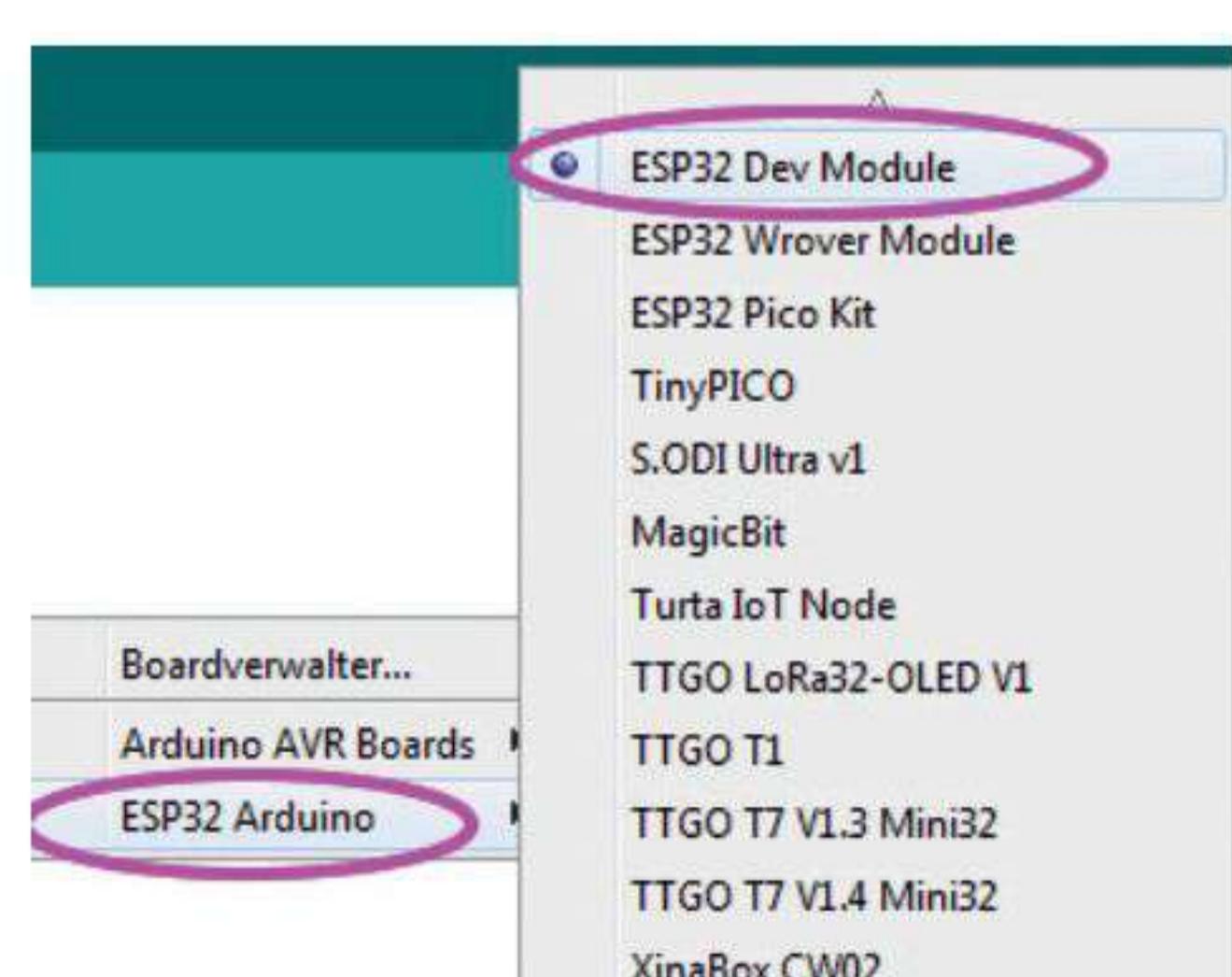
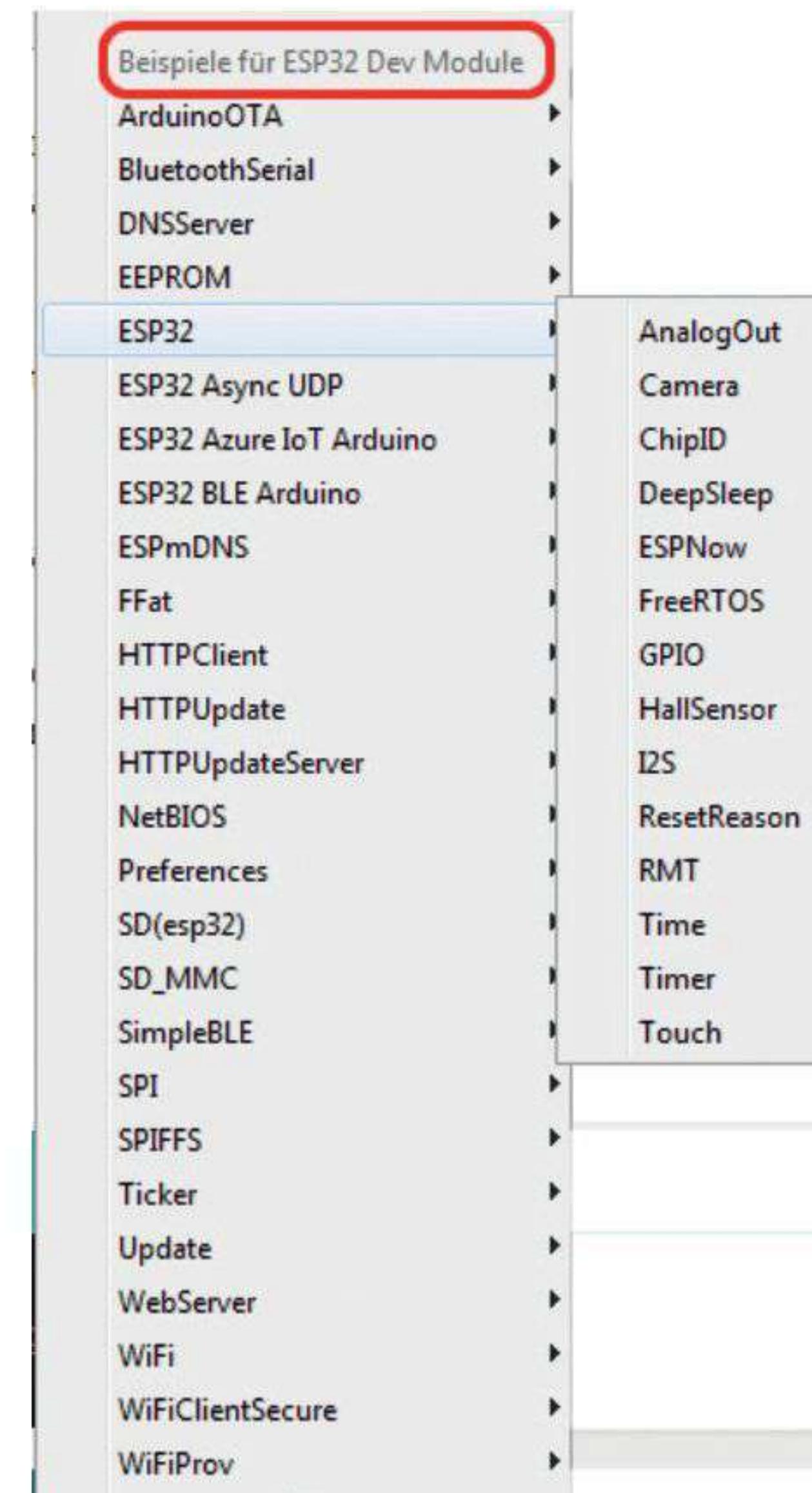


Bild 4: Nach der Installation des **ESP32-Pakets** sind unter **Board** die neuen Platinen wählbar. Zum hier benutzten Modul passt der rot markierte Eintrag.

Im *Boardverwalter*-Dialog tragen Sie als Suchbegriff *ESP32* ein und installieren die *ESP32-Boards*. Eventuell muss die *Arduino-IDE* neu gestartet werden, damit sie in der Liste sichtbar werden. Nutzen

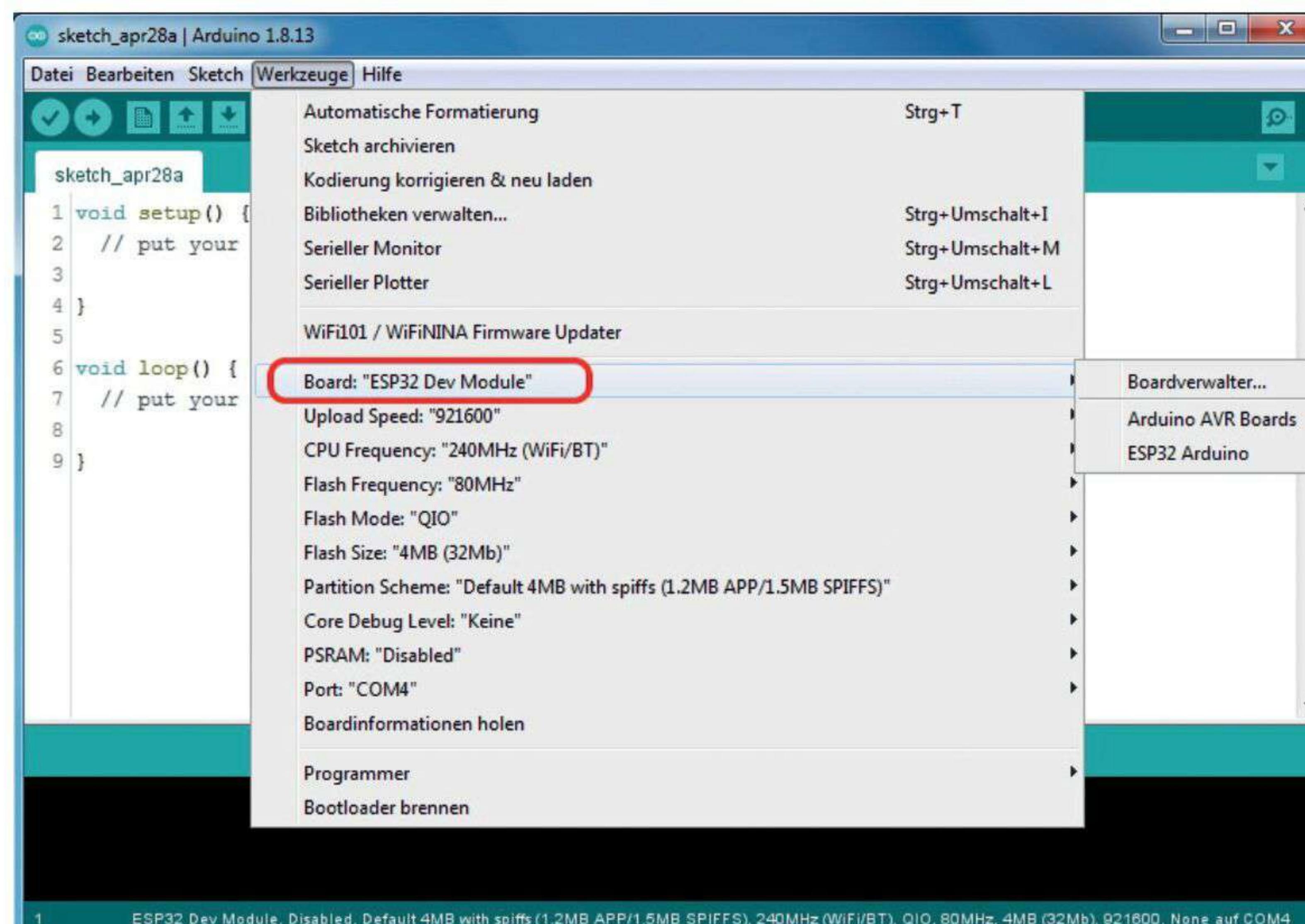


sich bereits auf der Platine, das Programm liest sich so:

```
#define LED_ONBOARD 2

void setup()
{
  pinMode(LED_ONBOARD, OUTPUT);
}

void loop()
{
  digitalWrite(LED_ONBOARD, HIGH);
  delay(1000);
  digitalWrite(LED_ONBOARD, LOW);
  delay(1000);
}
```



Erfahrene *Arduino*-Programmierer werden keinen Unterschied zu einem Sketch für einen *Arduino UNO* / *NANO* oder ähnliche Platinen mit 8-Bit-Controller feststellen. Das ist gut so, so kann der Nutzer auf bisherige Erfahrungen zurückgreifen. Um diese Zeilen ans Laufen zu bringen, müssen Sie den Sketch nicht abtippen. Alle Sketche finden Sie auf der virtuellen DVD im Ordner *\Artikel\ESP32*. Kopieren Sie die Sketch-Ordner auf Ihre Festplatte unter *Dokumente\Arduino*. Laden, kompilieren und übertragen Sie das Programm auf den *ESP32*. Die an *GPIO2* angeschlossene blaue LED sollte nun im Sekundentakt blinken. Bei anderen *ESP32*-Entwicklungsplatinen mag die LED evtl. an einem anderen *GPIO*-Pin angeschlossen sein. In diesem Fall ist der Wert in der `#define`-Zeile anzupassen.

Bild 5: Korrekte Einstellung für das Modul mit **ESP WROOM-32-Chip**.

NodeMCU32

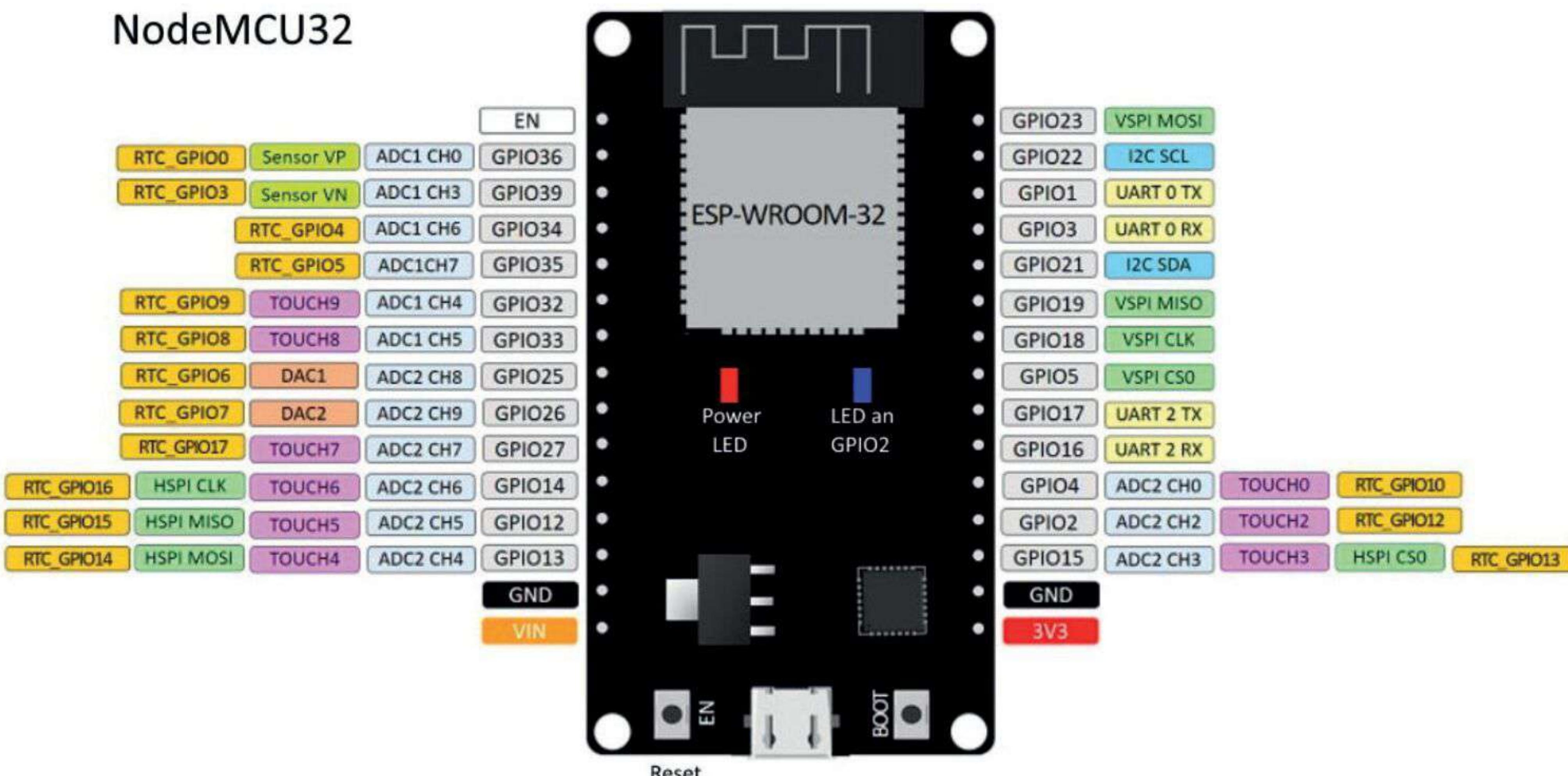


Bild 7: Pins und deren Bedeutung der NodeMCU32-Platine.



Bild 8: Der Sketch liest den Hal-Sensor des Controllers aus und zeigt Werte im seriellen Plotter der Arduino IDE. Den Sketch finden Sie auf der virtuellen DVD.

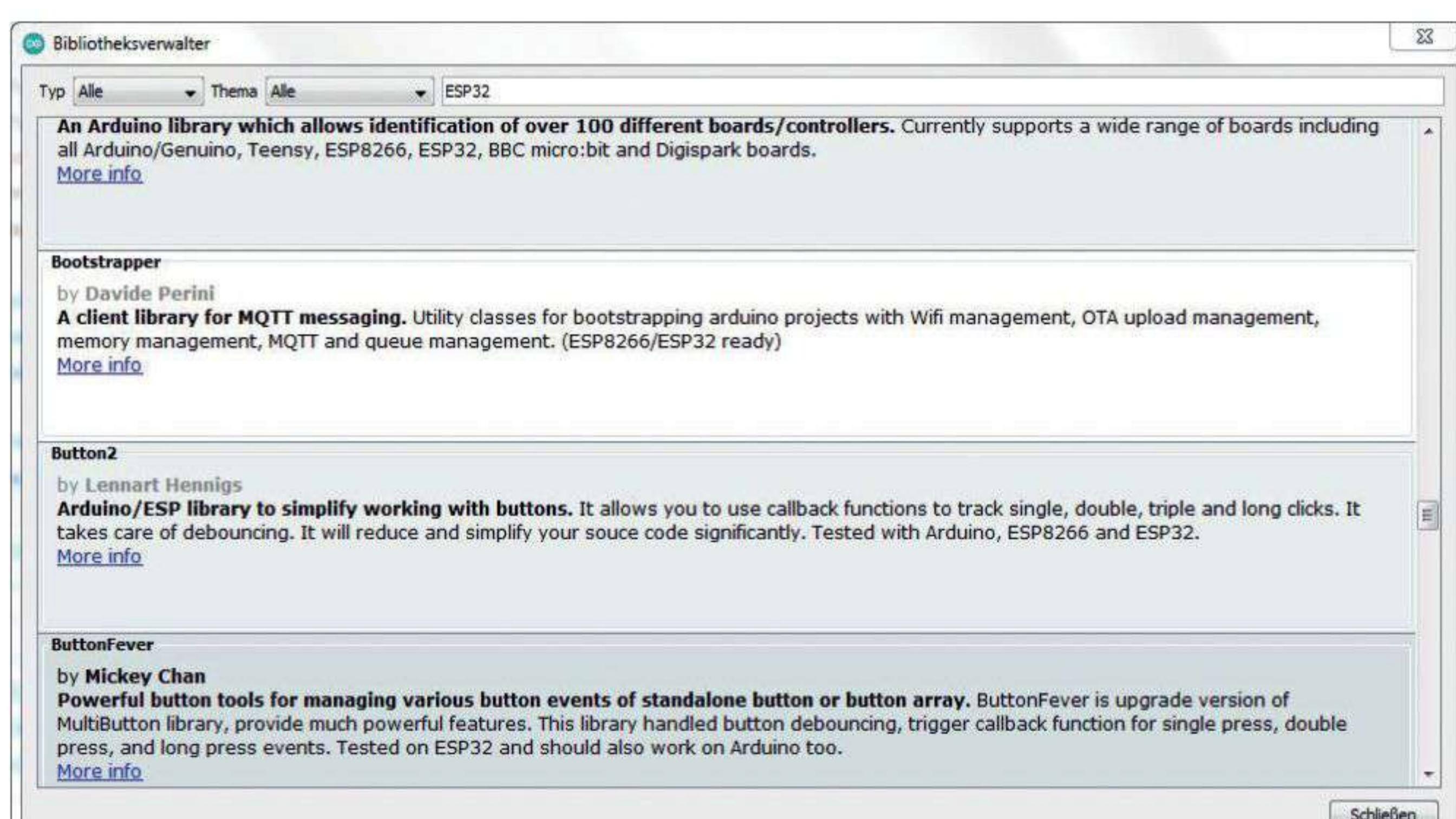
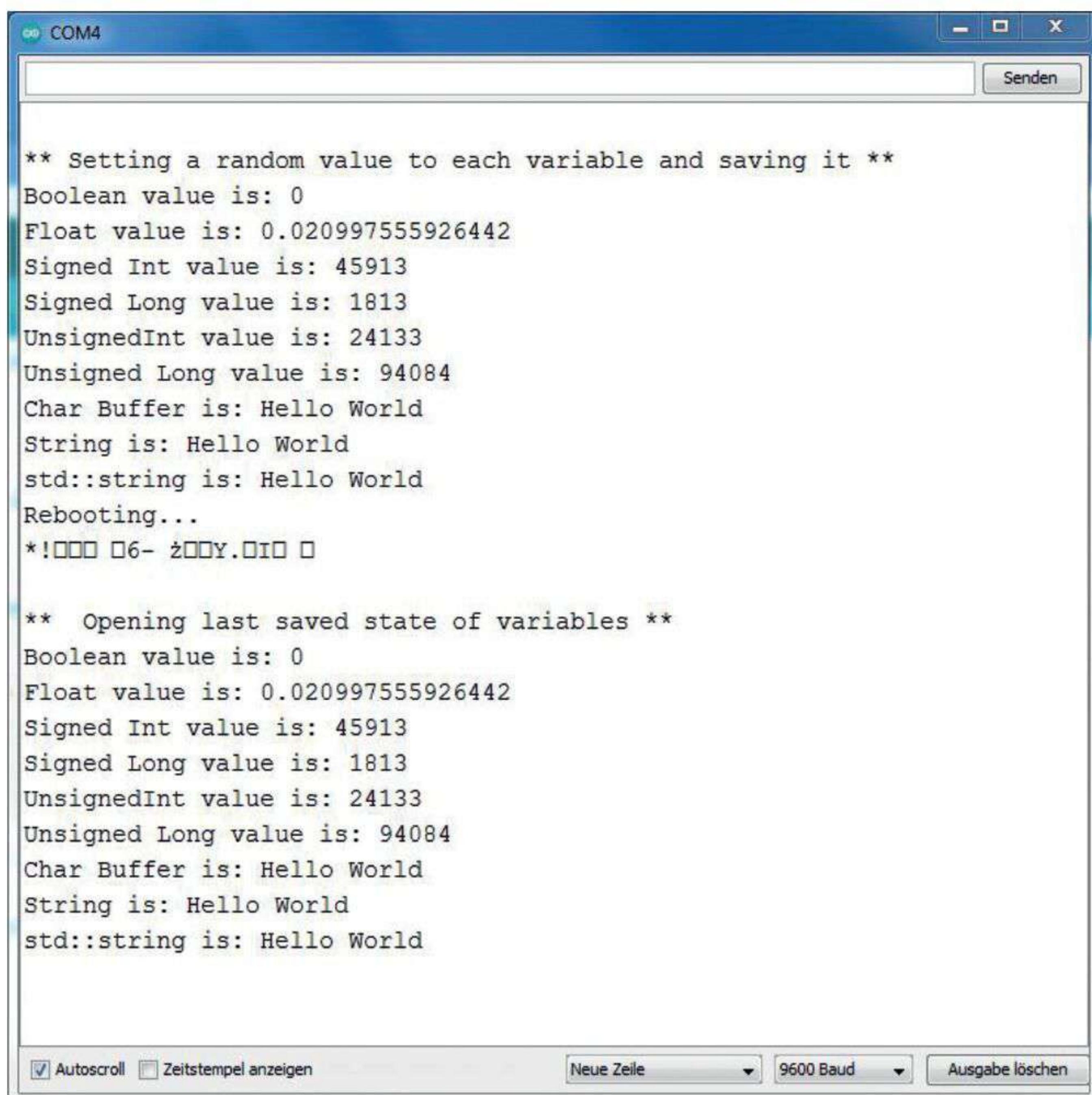


Bild 9: Gute Unterstützung bei der Programmierung: Der Bibliotheksmanager listet –zig Libraries für den ESP32 auf. Bevor man etwas programmiert, sollte man hier nach geeigneten Bibliotheken suchen. Mit jeder frisch installierten Library kommen dazu passende Beispiele in die Arduino IDE.

- Bild 9 zeigt einen Ausschnitt der für ESP32 angebotenen Programm-bibliotheken. In der folgenden Liste finden Sie eine Auswahl der interessantesten Libs:
- ACAN2515 u.a.: Mehrere CAN- Library für den Datenaustausch per Controller Area Network.
- bitluni's ESP32Lib: Das ist eine Sammlung von Funktionen für den ESP32, einschließlich

leistungsfähigster VGA-Grafik (Sprites, 3D), Sound und Game- Controller, verpackt in einer Arduino-Bibliothek.

- BlinkControl**: BlinkControl ist eine Arduino/ESP32-Bibliothek, die für die Verwaltung mehrerer blinkender LEDs mit verschiedenen Mustern entwickelt wurde. Kann mit digitalen Pins, analogen Pins oder 74HC595 Schieberegister arbeiten. Jede Instanz des Moduls steuert einen Pin. Verfügt über vordefinierte Blinkmuster. Sie können das Blinkmuster auch steuern, indem Sie ein Muster-Array bereitstellen. Vielleicht für eine Rufzeichen-LC-Anzeige?
- Arduino Bootstrapper**: und schreiben von Infos im JSON-Format, senden von Nachrichten an eine MQTT-Warteschlange, verbindet sich mit WiFi, ohne Passwörter hart zu kodieren.
- Button2**: Arduino/ESP32-Bibliothek zur Vereinfachung der Arbeit mit Tastern. Diese Bibliothek ermöglicht es Ihnen, Callback-Funktionen zu verwenden, um einfache, doppelte, dreifache und lange Klicks zu verfolgen. Sie kümmert sich um die Entprellung. Die Verwendung dieser Bibliothek wird Ihren Quellcode erheblich reduzieren und vereinfachen. Sie wurde mit ESP8266 und ESP32 getestet.
- APRS-IS**: Zugang zum APRS- Server und zu APRS-IS. Stellt eine einfache Schnittstelle zum APRS-Tier2-Server für ESP32- Boards bereit. Sie können die Bibliothek für ein APRS-iGate verwenden.
- Arduino OTA**: Mit dieser Bibliothek können Sie Sketches auf Ihrem Board über WiFi oder Ethernet aktualisieren!
- AsyncTelegram**: Ermöglicht das Senden von Mitteilungen an den Telegram-Dienst (gehört zu den sog. „sozialen Medien“).
- BH1750**: Unterstützt den Lichtsensor BH1750.
- DS323x**: Eine kostengünstige, extrem genaue I2C-Echtzeituhr (RTC) mit integriertem temperaturkompensiertem Quarzoszillator (TCXO) und Quarz. Der Baustein verfügt über einen Batterieeingang und behält die genaue Zeitmessung bei, wenn die Hauptstromversorgung des Bausteins unterbrochen wird.



```
** Setting a random value to each variable and saving it **
Boolean value is: 0
Float value is: 0.020997555926442
Signed Int value is: 45913
Signed Long value is: 1813
UnsignedInt value is: 24133
Unsigned Long value is: 94084
Char Buffer is: Hello World
String is: Hello World
std::string is: Hello World
Rebooting...
*!@00 06- 200Y.010 0

** Opening last saved state of variables **
Boolean value is: 0
Float value is: 0.020997555926442
Signed Int value is: 45913
Signed Long value is: 1813
UnsignedInt value is: 24133
Unsigned Long value is: 94084
Char Buffer is: Hello World
String is: Hello World
std::string is: Hello World
```

Bild 10: Der serielle Monitor zeigt das Speichern und Zurücklesen von Daten in das und aus dem SPI Flash File System (SPIFFS). Dabei werden die Daten im Flash-Speicher abgelegt, jedoch wie eine Datei angesprochen. Dieses Beispiel baut auf die Effortless-SPIFFS-Library.

- DS323x- Bibliothek: Eine Bibliothek für die DS3231/DS3232 extrem genaue RTC.
- Effortless SPIFFS: Eine Klasse, die das Lesen und Speichern von Daten auf dem ESP32 mühelos macht. Diese Bibliothek zielt darauf ab, den Zugriff auf SPIFFS (SPI Flash File System, das chipinterne Dateisystem) zu vereinfachen.
- EMailSender: ESP32 sendet E-Mail per WiFi, z. B. wenn am Controller ein Ereignis auftritt.

Das sind nur einige interessante Bibliotheken im Buchstabenbereich A bis E. Die Liste könnte man leicht bis zum Buchstaben Z weiterführen, was an dieser Stelle nicht sinnvoll erscheint. Dieser Abriss soll jedoch verdeutlichen, dass für sehr viele Anwendungen eine wertvolle Programmierunterstützung dank ausgetesteter Bibliotheken existiert. Im Internet gibt es viele Informationsquellen, wie die Literaturliste beweist. Weitere Links finden Sie auf der virtuellen DVD im Ordner Artikel\ESP32, dort gibt es weitere Bilder und eine größere Linkliste mit weiteren Informationen.

Und wann starten Sie Ihre Programmierung? Beispiel: Benutzen Sie die APRS-IS-Bibliothek, um von einem lokalen Funkgerät empfangene APRS-Meldungen in das Internet an einen APRS-Server zu leiten. Viel

Freude und Programmierspaß mit dem ESP WROOM-32.

ESP32 und der Amateurfunk

Sucht man im Internet nach „ESP32 decoder Arduino“, wird man an verschiedenen Stellen fündig: David, G6EJD, zeigt auf YouTube unter anderem seinen CW-Decoder mit OLED-Display. Das Video zeigt eine fehlerfreie Dekodierung (bei klarem Empfang ohne QRN etc.) Er ist ein fleißiger Bastler, das Anschauen seiner Videos lohnt, und immer ist ein ESP32 oder ein (älterer und kleinerer) ESP8266 mit an Bord. Einige seiner Programme findet man auf Github [6]. Einen Encoder-Lib für die Modes JT65, JT9, JT4, FT8, WSPR und FSQ für Arduino und ESP32 bietet User „Etherkit“ ebenfalls auf Github an [7]. Bei der Programmierung haben die bisher unerwähnt gebliebenen DSP-Funktionen des ESP32 sicherlich geholfen. Programmierer, die sich für die Datenverarbeitung (Dataprocessing) unter Arduino (alle Platinen) interessieren, schauen in die beachtlich lange Linkliste bei [8] herein. Rund um APRS wurden von mehreren OMs – da es die APRS-Lib für Arduino gibt – ein APRS-Gate, ein TNC und ähnliches realisiert.

Ein guter Platz zum Start bieten die Webseiten und Videos von Klaus, DJ7OO [9] oder [10].

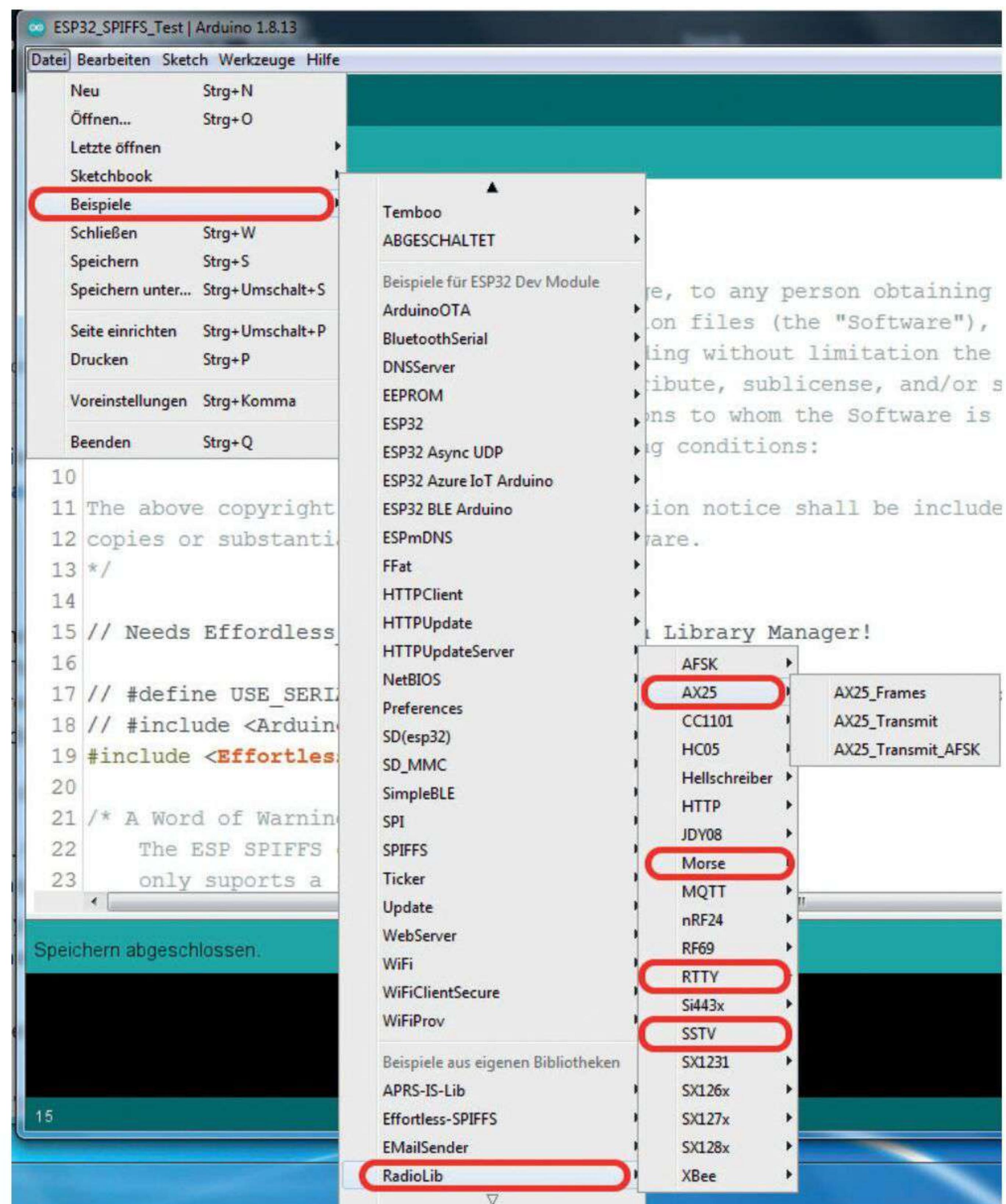


Bild 11: Nach der Installation der RadioLib von J. Gromes – vermutlich ein Funkamateuer – finden sich Beispielprogramme für einige Afu-Betriebsarten, die zum Experimentieren einladen.

Eine besondere Beachtung sollte die RadioLib von J. Gromes erfahren, die für verschiedene Arduinos und für den ESP32 entwickelt wurde. Diese Lib unterstützt ein sog. Radio-Shield, aber auch eine Reihe interessanter und beliebter Transceiver-Funkmodule [11,12] wie das HopeRF-RF69 oder die Lora-Module der Reihe SX127x/RF-M9x und viele mehr. Die RadioLib installiert man wie üblich mit dem Bibliotheksmanager. Danach findet man eine Menge neuer Beispiel-Sketchs (Bild 11) in der Arduino IDE. Unter anderen befinden sich interessante Beispiele zum Senden von AX25-Datenpakete (für APRS), zum Senden von Morsezeichen in AFSK, RTTY und SSTV. Hier ergibt sich ein schönes Betätigungsfeld für den Funkamateuer.

Literatur/Verweise:

- [1] Github Espressif: <https://github.com/espressif>
- [2] Chipinfo: <http://esp32.net/>
- [3] ESP32-Projekte: <https://diyprojects.io/mcu-esp32-esp8266-stm32-arduino/esp32-iot/>
- [4] E-Book (PDF) der TU-Berlin: https://www.user.tu-berlin.de/liecke/dEIn_Labor/Smart%20Home/ESP32/das-offizielle-esp32-handbuch-ebook.pdf
- [5] ESP32-Wiki: <https://desire.giesecke.tk/index.php/2018/01/30/esp32-wiki-entries/>
- [6] G6EJD-Programme auf Github: <https://github.com/G6EJD>
- [7] „Etherkits“ Software bei Github: <https://github.com/etherkit/JTEncode>
- [8] Arduino Data Processing-Liste: <https://www.arduino.cc/reference/en/libraries/category/data-processing/>
- [9] Webseite von DJ7OO: <http://www.kh-gps.de/>
- [10] LoRa, APRS und ESp32: <https://www.aronaut.at/2019/12/lora-aprs-tracker-mit-ttgo-tbeam-433mhz/>
- [11] RadioLib von J. Gromes: <https://github.com/jgromes/RadioLib>
- [12] Radio-Lib, Liste der Funkmodule: <https://github.com/jgromes/RadioLib/wiki/Modules>

Start mit dem Arduino UNO

Liegt ein Arduino UNO unbenutzt in der Grabbelkiste, so sollte man die Platine zum Leben erwecken. Dann kommen Sie für Null Euro und in nur wenigen Minuten zu einem Hobby-Logikanalysator! Mit sechs Kanälen und bis zu vier MHz Abtastrate schafft der Kleine immerhin 1024 Samples. Das ist ausreichend für vielerlei alltägliche Aufgaben, solange die zu messenden Signale nicht zu schnell daherkommen. Auf [1] findet man das Programm für den Arduino UNO, aber auch auf der virtuellen DVD zum Download. Schauen Sie in den Ordner *Artikel\Arduino_LA*. Kopieren Sie das Verzeichnis *logic_analyzer* in Ihr Projektverzeichnis der Arduino-IDE (z.B. Dokumente\Arduino). Laden Sie den Sketch *logic_analyzer.ino*, stellen in der Arduino-IDE (Menü *Werkzeuge*) das Board und den Com-Port ein und übertragen schließlich den Sketch auf den Arduino UNO. Das ist geschafft. Sollte es dabei Probleme geben, schauen Sie bei einer Videoplattform nach Tutorials der Art „Erste Schritte mit Arduino“. Was Sie noch wissen müssen: Es stehen sechs Eingangskanäle zur Verfügung. Kanal 0 ist mit dem Arduino Pin 8 verbunden, Kanal 1 mit Arduino Pin 9 und so weiter bis Pin 12. Beachten Sie bitte: Da es sich um digitale Eingänge des Prozessors handelt, darf die angelegte Spannung max. 5 Volt betragen! Höhere Spannungen müssen Sie mit einem Spannungsteiler oder Level Shifter auf einen Bereich zwischen 3,3 und 5 Volt verringern.

... und jetzt mit dem Arduino Mega 2560

Und nun kommt der große Bruder ins Spiel, der Arduino MEGA! Sie benutzen denselben Arduino-Sketch, kompilieren dieses Mal für den Arduino MEGA2560. Besitzen Sie eine ältere Version der Arduino IDE, wählen Sie als Board *Arduino MEGA* aus. Vorteil: Statt auf 1024 Samples beschränkt zu sein, nutzen Sie den größeren Speicher des Controllers und kommen insgesamt auf immerhin 7 * 1024 Bytes und statt sechs auf acht Kanäle. Auf der MEGA-Platine

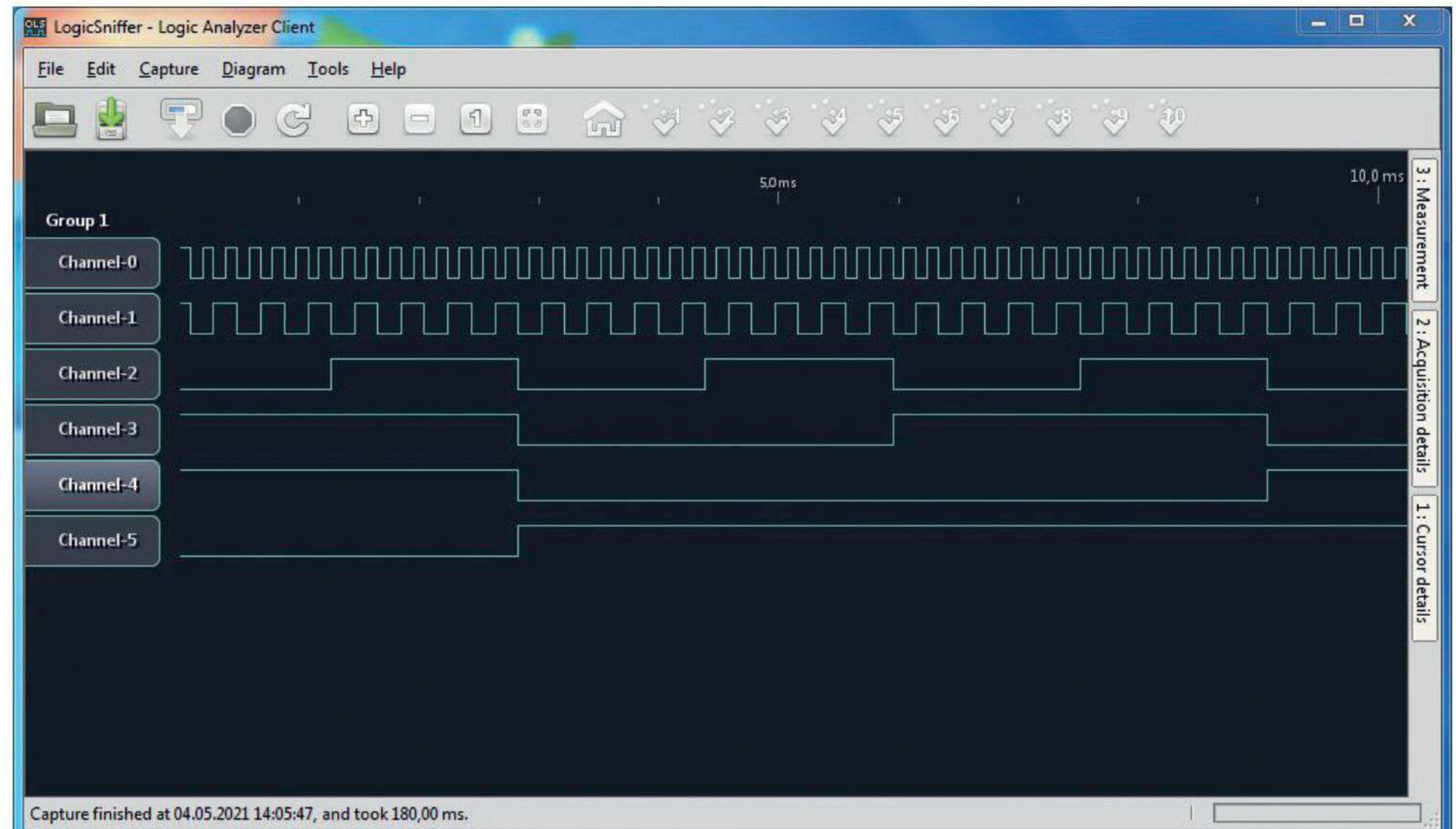


Bild 1: Digitale Zeitmessungen mit Arduino und Logic Sniffer.

Logikanalysator für das Hobby: Arduino als digitaler Schnüffler

Wer digitaler Elektronik entwickelt oder nachbaut, kommt bald an den Punkt, das zeitliche Verhalten eines oder mehrerer digitaler Signale im Verhältnis zu anderen zu beurteilen. Das, was das Oszilloskop für analoge Signale leistet, ist etwa der Logikanalysator für die digitale Welt. Für das Elektronikhobby braucht es dazu kein kostspieliges Messgerät, in diesem Fall reicht ein Arduino und die „richtige“ Software aus. Selbst für wenige Messungen ist es reizvoll, sich in wenigen Minuten einen Logikanalysator (LA) zusammenzustecken.

ist Kanal 0 auf dem Arduino-Pin 22 zu finden, Kanal 1 auf Pin 23 und so fort, bis Kanal 8 auf Pin 29 zu liegen kommt. Nach diesen Vorbereitungen steht also ein Arduino UNO oder MEGA als Logikanalysator (LA) zur Verfügung. Aber wie diesen bedienen?

Bild 2: Vor der Messung: Geräteeinstellungen für den Arduino UNO. COM-Port bitte anpassen!

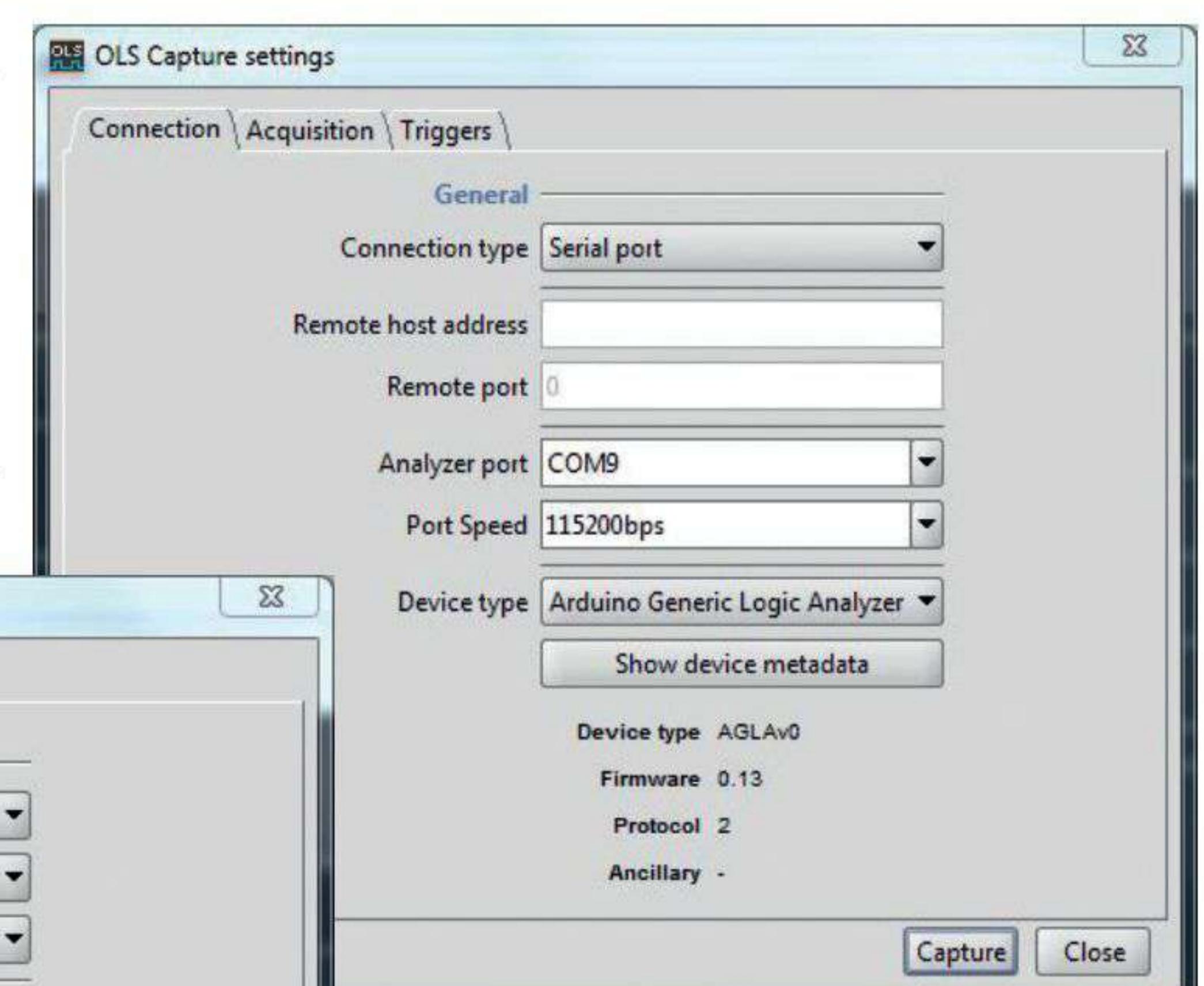
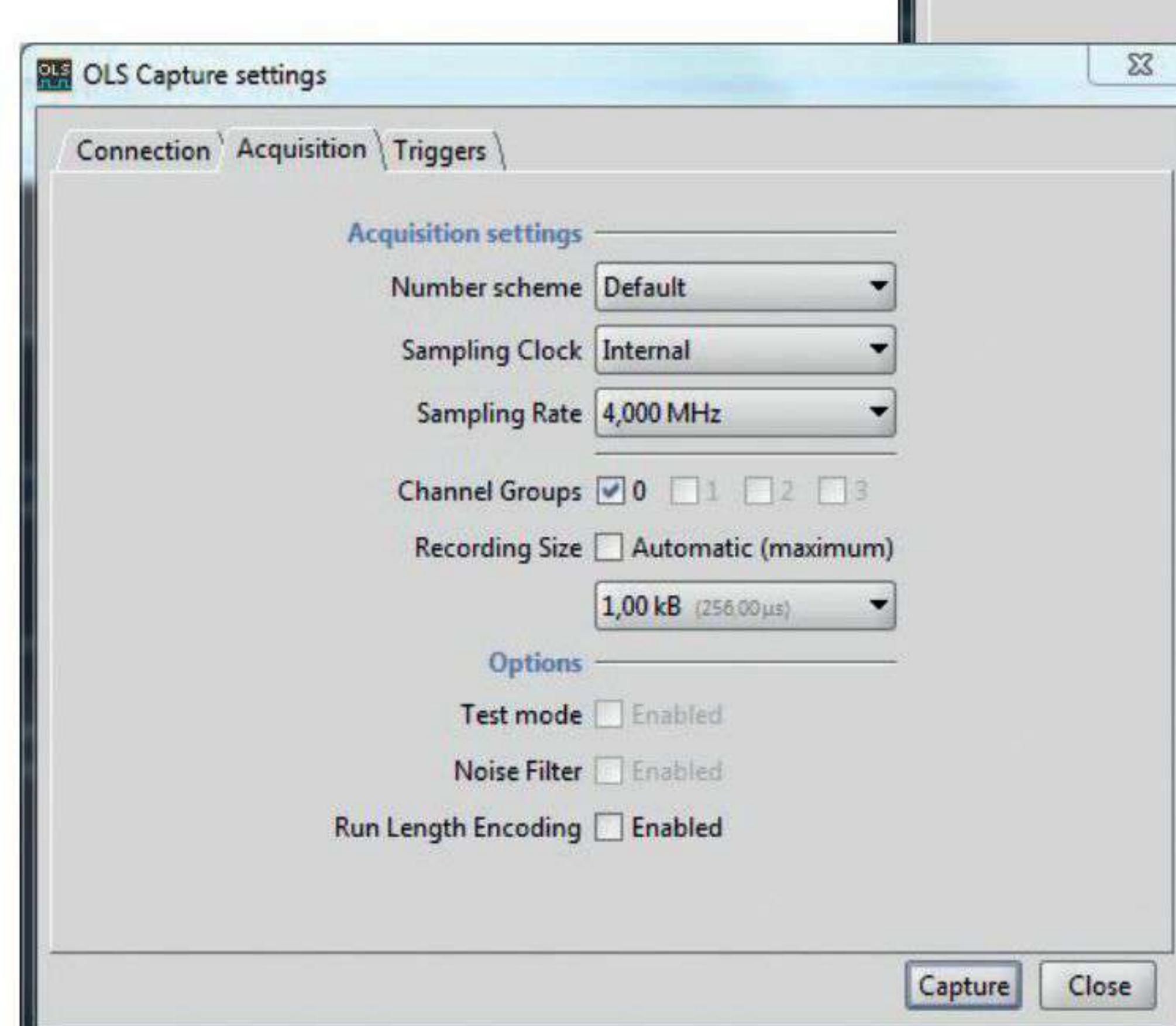


Bild 3: Wichtige Einstellungen von Abtastrate und Speichergröße (Anzahl Samples).



Logic Sniffer als „virtuelles Messgerät“

Für die Arbeit mit Ihrem Logikanalysator fehlt noch ein Programm zur Abfrage und Anzeige der Messdaten. Dazu eignet sich prima der

kostenfreie Logic Sniffer [2]. Es benötigt Java [3], das auf den meisten Windows-PC bereits installiert ist. Auf Linux und MacOS wird der Schnüffler ebenso lauffähig sein, das wurde jedoch nicht getestet. Man startet das Programm unter Windows mit *run.bat*. Sie können es direkt von der virtuellen DVD starten.

Das Aufmacherbild (Bild 1) zeigt, wie es ausschaut. Für die Graphen habe ich einen Zähler an den Arduino UNO angeschlossen. Von Kanal zu Kanal verringert sich damit die Frequenz des Rechtecksignals auf die Hälfte. Vor jeder Messung werden der LA (das Gerät) und deren Parameter in einem Dialog festgelegt. Im Menü *Capture* wählen Sie *Begin Capture*. Es erscheint der Dialog lt. Bild 2. Wählen Sie nun den COM-Port des Arduinos, stellen die Baudrate auf 115200 Baud und wählen bei einem Arduino UNO unter *Type* den Eintrag *Arduino General Logic Analyzer*, bei dem Arduino Mega aber *Arduino Mega Logic Analyzer*. Mit einem Klick auf den Button *Show device Metadata* erfahren Sie, ob das Programm ihren Arduino-LA erkennt. Im Bild 3 sehen Sie, welche Parameter man im Reiter *Aquisition* einstellen kann. Das sind insbesondere die Abtastrate (sampling rate) und die Anzahl der Messungen (recording size). Weitere aufgeführte Parameter sind für diese Hardware nicht relevant.

Einfach triggern

Manchmal ist es schwierig, in einem digitalen Signal den richtigen Zeitpunkt zum Start der Messung zu finden. Für diesen Fall sind Trigger sehr hilfreich. Bild 4 zeigt Triggeereinstellungen. Ist bei der Maske ein Häkchen gesetzt, bedeutet das,

dass dieser Kanal auf einem Low-to-High-Wechsel wartet, aber nur, wenn beim Wert zum Kanal auch ein Haken gesetzt wurde. Im Bild wurde der Trigger so eingestellt, dass die Messung beginnt, wenn Kanal 0 und Kanal 3 zugleich High-Pegel zeigen. Entfernt man in der Maske aber das Häkchen bei Kanal 0 und 3, startet die Messung erst, sobald beide Kanäle Low-Pegel aufweisen. Probieren Sie es aus!

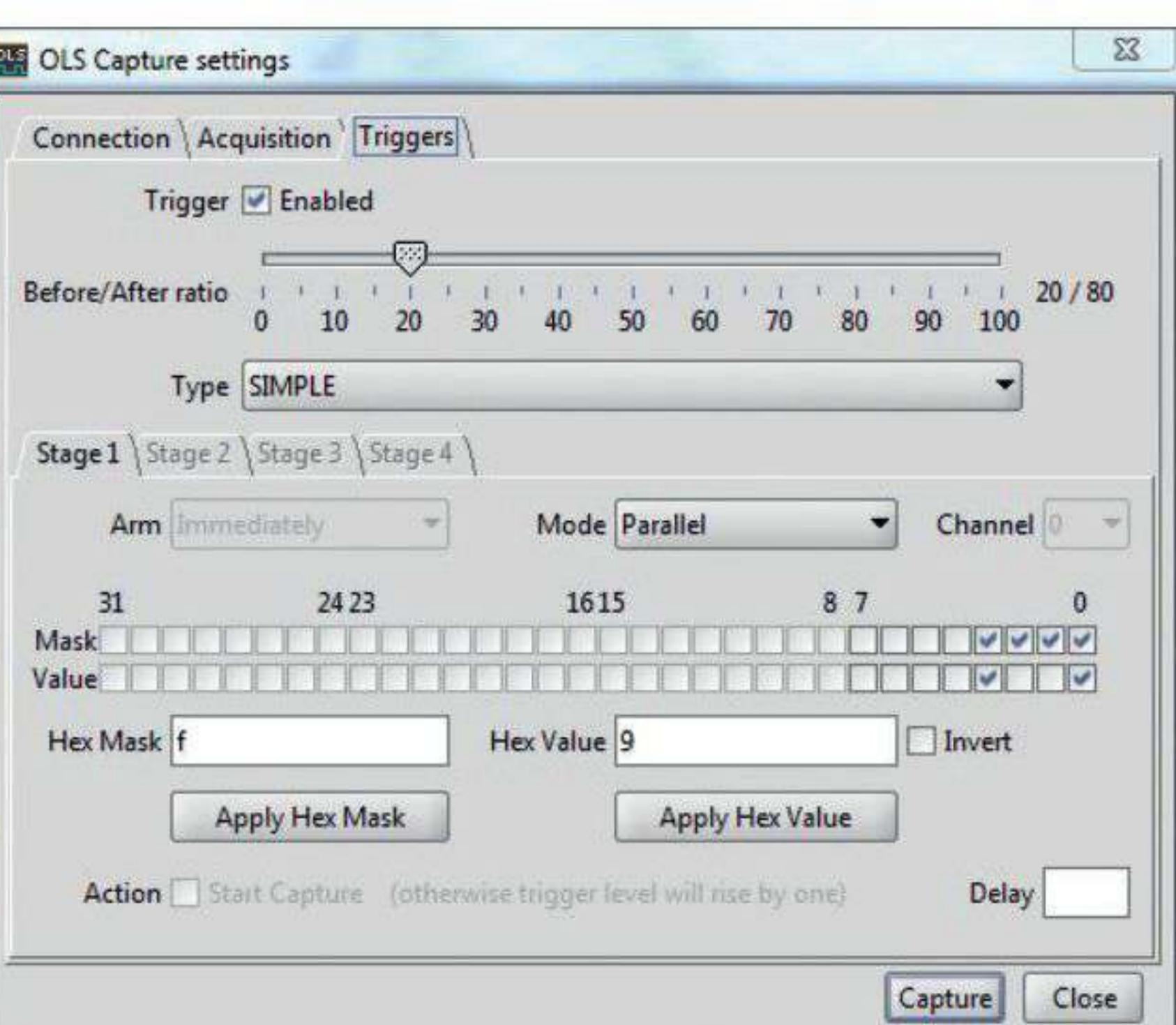
Die erste Messung starten Sie dann mit einem Klick auf den Capture-Button. Weitere Messungen mit denselben Parametern starten Sie mit STRG+R oder im Menü *Capture* mit *Repeat Capture*.

Logic Sniffer wäre kein „Messgerät“, könnte man Resultate nicht ausmessen. Im Messmodus (*Measure-Mode*), den man z.B. im *Tools*-Menü aktiviert, misst das Programm an der Mausposition jeweils die Zeit einer Periode aus. Mit der rechten Maustaste setzt man senkrechte Marker (hier als Cursor bezeichnet), die man, mit der Maus „anfasst“ und nach dem Setzen falls nötig verschiebt. Bild 5 zeigt eine Messsituation: Es wurden zwei Marker 1 und 2 gesetzt. Rechts im Bild ist der ausgeklappte Infobereich des Reiters *Measurement* zu sehen, darin werden aus der Stellung der Marker zueinander Zeitabstand, Frequenz und Einschaltzeit in Prozent (*duty cycle*) berechnet. Der Reiter *Aquisition details* gibt Auskunft über die Abtastrate, Abtastzeit, eine Angabe zur horizontalen Auflösung jedes Pixels und weiteres.

Der Reiter *Cursor details* schließlich zeigt die Zeitposition der Marker, eine Information, die man ebenso gut an den Markern selbst abliest.

Ein guter LA (und schon fast jedes moderne Oszilloskop) besitzt die Fähigkeit, diverse Protokolle zu dekodieren. Bild 6 zeigt das Ergebnis einer I2C-Analyse. Im Menü Tools stehen Dekoder für UART, SPI, für das 1-Wire-Protokoll und weitere

Bild 4: Optional kann man hier einfache, aber wirkungsvolle Trigger setzen.



Auf einen Blick

Arduino	Anz. Kanäle	Arduino-Pins	Samples	Typ:	Baudrate
UNO	6	8-12	1024	Arduino Generic Logic Analyzer	115200
MEGA 2560	8	22-27	7168	Arduino MEGA Logic Analyzer	115200

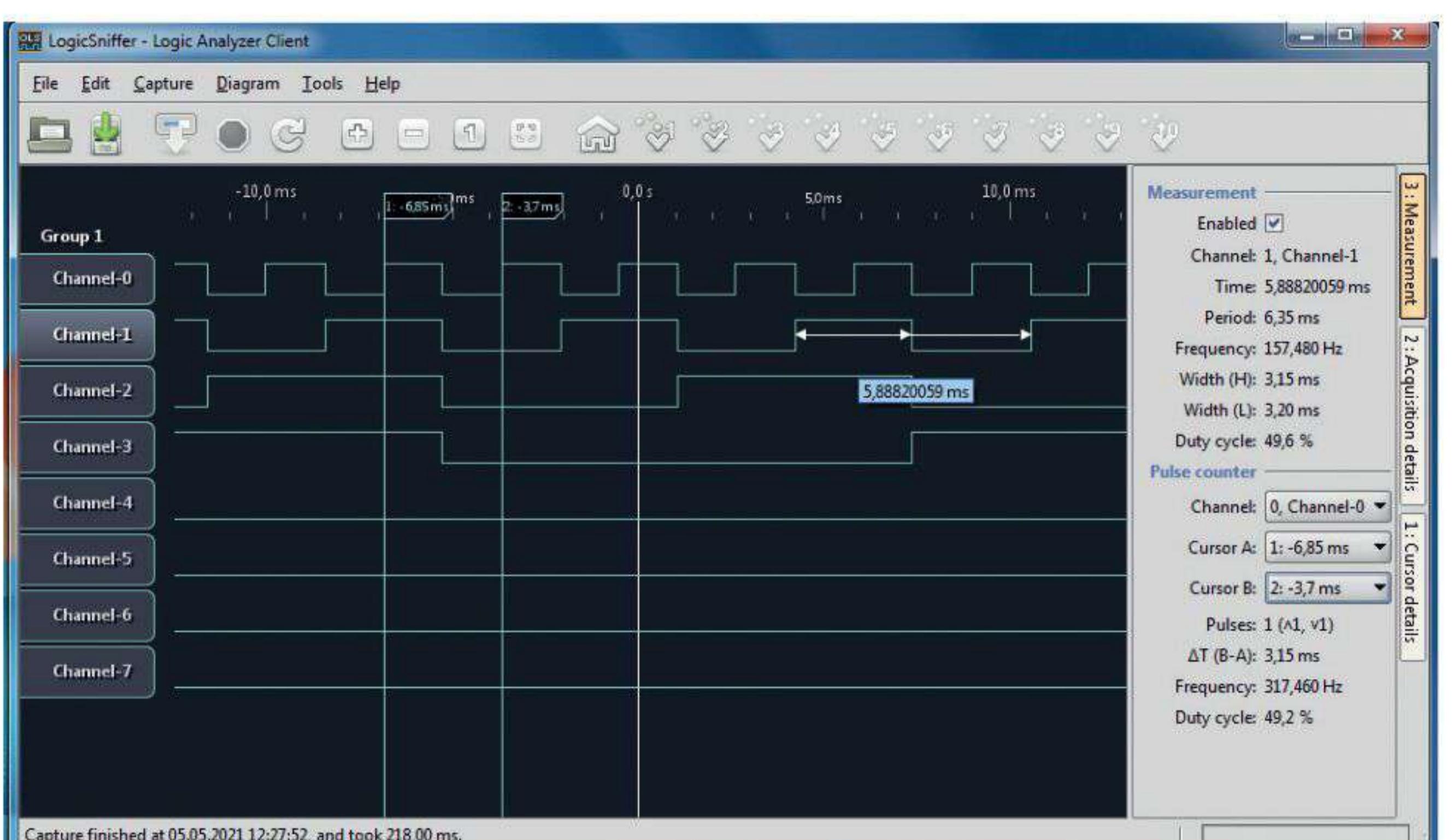


Bild 5: Logic Sniffer im Messmodus: Zwei Marker und das Infofenster rechts.

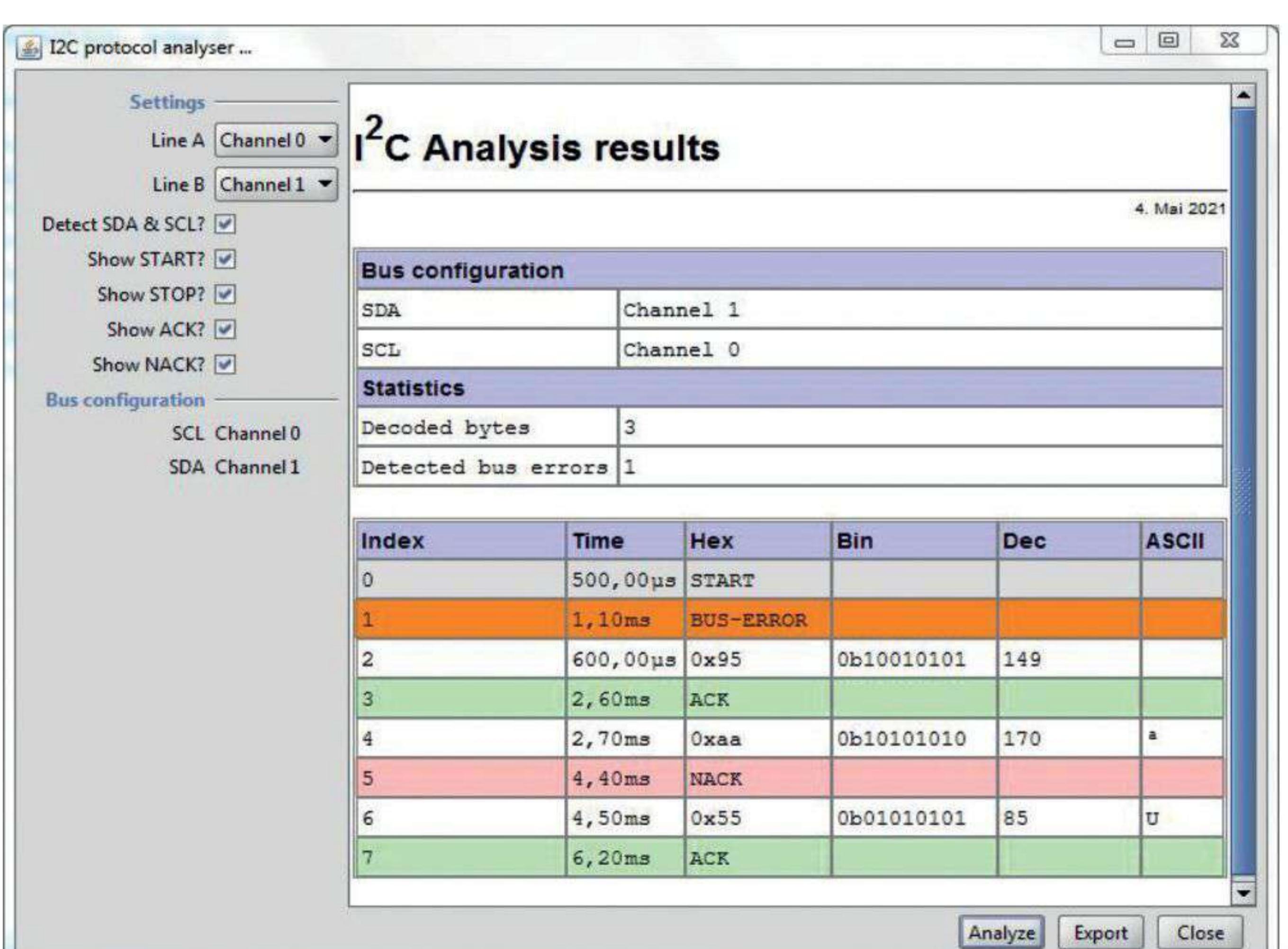


Bild 6: Logic Sniffer dekodiert verschiedene Protokolle, hier eine I2C-Analyse.

zur Verfügung. CAN fehlt, ist aber für Funkamateure weitgehend uninteressant. Programmierer werden sich über den JTAG-Dekoder freuen. Der Line-Decoder betrachtet einen Kanal und interpretiert nach der Manchester- oder Non Return to Zero-Methode. Für die Spionage auf einem (freien) COM-Port verfügt der Schnüffler über ein einfaches serielles Terminal, hier als *Serial Console* bezeichnet. Leider kann man damit nicht (ohne weitere Software) den COM-Port zum Arduino ausspähen.

Sicherlich ersetzt der (fast) Null-Euro-LA mit dieser Software keinen professionellen Logikanalysator, das wird vermutlich niemand erwarten.

Diese Kombination aus Arduino und Logic Sniffer ist jedoch weit mehr als „besser als nichts“: Mit sechs bzw. acht Kanälen, der maximalen Abtastrate von 4 MHz und den Möglichkeiten zur Messung und Dekodierung kann man allerlei Sinnvolles anstellen.

Literatur/Verweise:

- [1] LA mit Arduino auf Github: https://github.com/gillham/logic_analyzer
- [2] Logic Sniffer Java Sourcen: <https://www.lxtreme.nl/ols/>
- [3] Java zum Download: <https://www.java.com/de/download/manual.jsp>

Michael Wöste, DL1DMW

FT8 hat viele Vorteile: Geringe Sendeleistung, kleine Antennenanlagen, das Logbuch wird automatisch geführt und auf Fremdsprachenkenntnisse kann man verzichten. Doch oft werde ich von Anfängern dieser Betriebsart (besser: dieses Protokolls) gefragt: „Wieso kann ich nichts dekodieren?“ oder „Wieso gelingt mir kein QSO?“ und „Was kann ich tun, um QRV zu werden?“. Einige Tipps aus der Praxis.

Tipp 1: Zeit

Wichtig ist zunächst eine korrekte Installation und Konfiguration und – ganz wichtig – die Nutzung der aktuellsten WSJT-X Software. Immer wieder gibt es neue Versionen, manchmal wird das Datenformat oder das Zeitraster verändert oder angepasst. Ältere Versionen von WSJT-X funktionieren dann nicht mehr! Es ist weit bekannt, dennoch sei es hier erwähnt: Die WSJT-X-Software erfordert eine enge Zeitsynchronisation Ihres Computers. Das bedeutet, dass der Betrieb in FT8/FT4 in einem festen Zeitschema erfolgt. Dazu muss die Systemzeit des steuernden PC auf unter einer Sekunde genau korrekt eingestellt sein. Normalerweise ist die eingebaute Zeitsynchronisation in Microsoft Windows-Betriebssystemen nicht immer ausreichend. Um eine Abweichung von weniger als einer halben Sekunde zu erreichen, sollte jeweils vor dem Funkbetrieb die Systemzeit des Windows-Betriebssystems korrekt eingestellt werden. Windows 10 bringt dazu bordeigene Mittel mit, bei älteren Windows-Versionen empfehlen sich die Programme Meinberg NTP [2], NetTime oder für ältere PC (Windows XP etc.) Dimension 4 [3]. Meinberg NTP für Windows

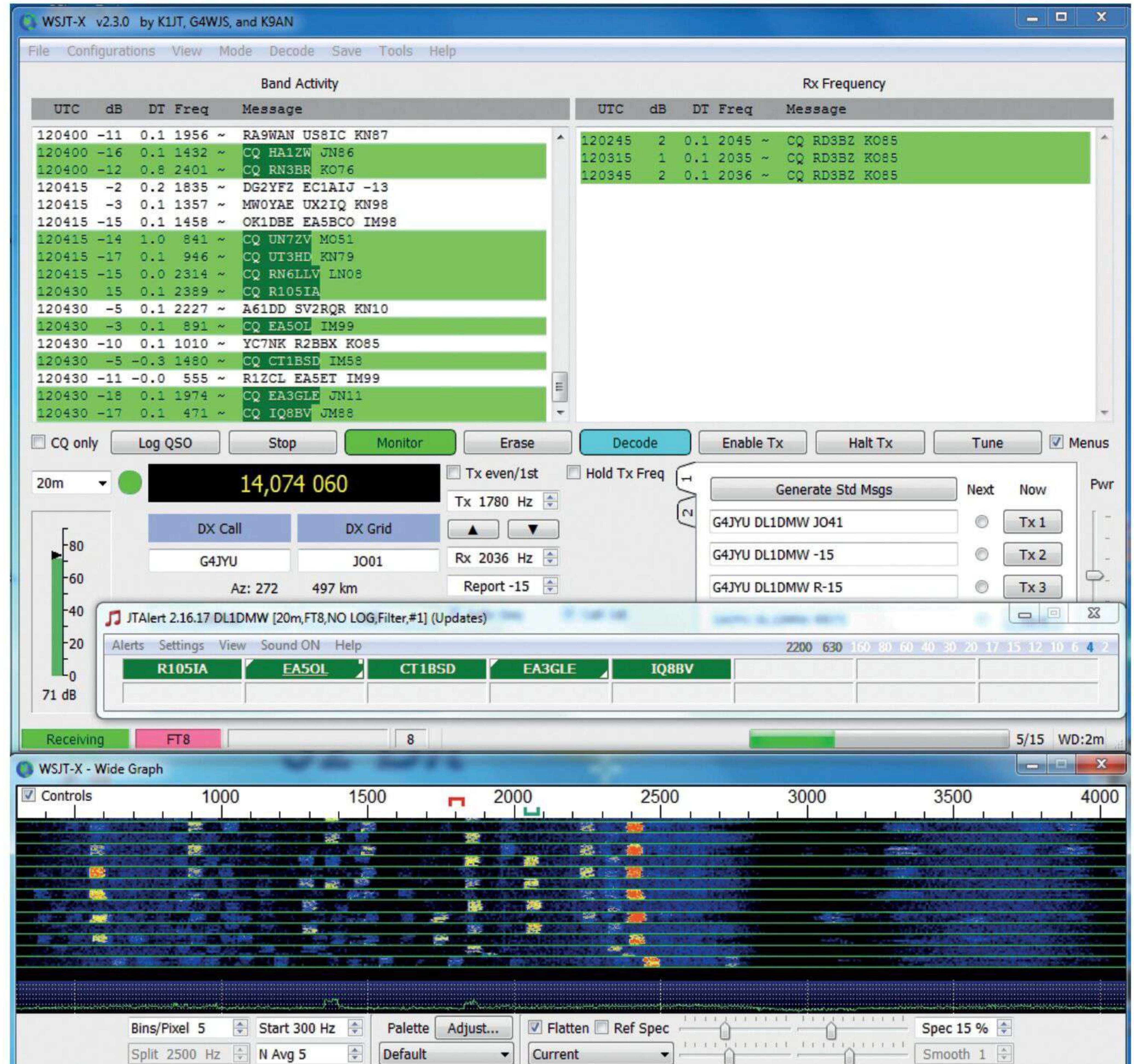


Bild 1: WSJT-X und JTAlert in Aktion.

FT8 und FT4

Tipps zum erfolgreichen Betrieb

installiert einen Windows-Service, der dafür sorgt, dass die Systemzeit automatisch aktualisiert wird. Das Programm ist Open Source und kostenlos. Ebenso kostenlos sind NetTime und Dimension 4, beides

Freeware, die man bei Bedarf startet, um die Systemzeit zu aktualisieren.

Im Fieldday oder, wenn Sie länger als ein oder zwei Tage ohne Internetzugang sind, empfiehlt sich ein anderer Weg zur Einstel-

lung der PC-Systemzeit, etwa via dem Zeitzeichensender DCF77 oder per GPS. Derlei Geräte kann man entweder selbst bauen oder im Handel erwerben.

Tipp 2: Die richtige Soundkarte

Es ist wichtig, dass Ihre Soundkarte des Funkgerätes oder des PC-Interfaces nicht als Windows-Standard-Soundkarte eingestellt ist. Wäre das so, könnten Windows-Systemklänge über den Sender gehen. Sind mehrere Soundkarten im PC installiert, prüfen Sie, dass in WSJT-X keine „falsche“ Soundkarte angewählt ist. WSJT-X benötigt eine Einstellung der Soundkarte von 16 Bit, 48000 Hz (DVD-Qualität). Schalten Sie eine mögliche Bassverstärkung oder ähnliches aus.

Häufig treten Probleme auf, nachdem Windows ein Software-

Band Activity				
UTC	dB	DT	Freq	Message
153115	-11	0.1	1568	~ CQ 9M2TDX OJ15
153115	-8	0.2	870	~ EA3NA SX8A 73
153115	-17	0.1	707	~ CQ KB1MGI FN42
153115	-19	0.1	1348	~ WXOMIK W3JAW FN10
153115	-19	0.5	1116	~ CT1BFP IZ6OYU RR73
153115	-5	0.3	1887	~ G3YBO K9US 73
153115	-18	0.1	1483	~ C6AJB KJ1R FN32
153115	-7	0.1	1876	~ CQ TF3MH HP94
153115	-12	1.0	879	~ GM0GDD RK4PR -16
153115	-15	0.2	791	~ DK8FG W7MY DN06
153115	-18	0.2	1732	~ LA5DSA KA0BOJ RR73
153115	-13	0.2	1901	~ CQ DX DL5AN JN49
153115	-16	0.3	2085	~ M10OBC KF0CPA EN17
153130	-5	0.1	979	~ IZ6OYU CT1BFP 73
153130	10	0.1	1276	~ LA2RT SV1EJD KM18
153130	-6	-0.0	1409	~ ON4VT RD3TBQ -09
153130	-6	0.1	2511	~ K9US US2IC -14

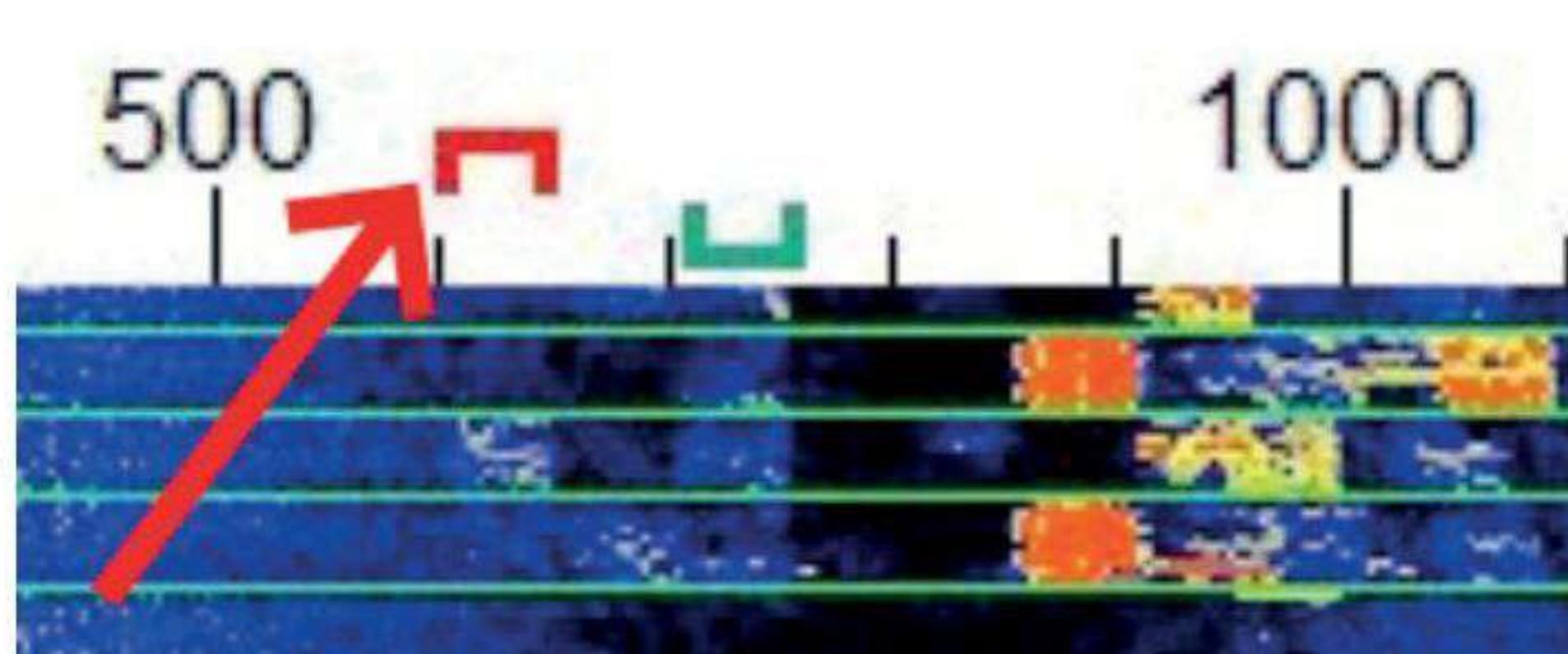
Bild 2: Die Spalte DT zeigt die Zeitabweichung. Aktualisieren Sie die Zeitsynchronisation der PC-Uhr, wenn bei mehreren Stationen höhere Werte als 1 Sekunde rot angezeigt werden.

Update ausgeführt hat oder Sie die WSJT-X-Software starten, bevor Ihr Radio eingeschaltet ist oder die USB-Kabel nicht eingesteckt sind. Ist das Funkgerät nicht über ein Audiokabel mit dem Eingang der Soundkarte des PC verbunden, schaltet Windows diesen Eingang ab. Er wird erst nach Einsticken des Audiokabels in den Eingang wieder aktiv. Bevor Sie WSJT-X starten, sollten alle Verbindungen zwischen Funkgerät und Computer vorhanden sein. Je nach verwendetem Funkgerät müssen Sie möglicherweise Änderungen an dessen Einstellungen vornehmen. Dazu gibt es Anleitungen zum Einrichten bestimmter Funkgeräte für WSJT-X im Internet oder als Videos auf YouTube. In den Einstellungen des Funkgerätes sollten Sie überprüfen, ob die NF-Bandbreite auf die maximale Einstellung eingestellt ist, sofern das Gerät dies unterstützt. Nachdem WSJT-X korrekt konfiguriert ist und die PC-Systemzeit synchronisiert ist, ist der nächste wichtige Punkt die Wahl der Sendefrequenz, um sicherzustellen, dass andere Stationen Sie hören können.

Tipp 3: Die freie Frequenz

Wählen Sie zum Senden immer eine freie Frequenz! Im Gegensatz zu CW oder SSB, wo es wichtig ist, dass die Sendefrequenz mit der der Gegenstation übereinstimmt, sollten Sie nicht versuchen, FT8- und FT4-Stationen auf deren Sendefrequenz anzufragen. Verwenden Sie keine belegte Frequenz! Da jede Station alle Signale im Durchlassbereich der ZF dekodiert, ist es egal,

Bild 3: Einstellung von Sende- und Empfangsfrequenz: Klick links bestimmt die Empfangsfrequenz, Shift + Klick links setzt die Sendefrequenz. Verwenden Sie immer eine freie Frequenz für die Aussendungen.



auf welcher Frequenz Sie rufen. Wählen Sie daher immer eine freie Frequenz aus, auf der keine andere Station sendet. Dann werden Sie gehört. Zur Einstellung der Sendefrequenz können Sie Ihre Maus in dem WSJT-X-Wasserfall nutzen. Mit Shift + linke Maustaste stellt man die Frequenz zum Senden ein, siehe Bild 3. Die rote Klammer markiert die Sendefrequenz, die grüne die Empfangsfrequenz. Die grüne Klammer setzen Sie mit einem linken Mausklick. Die Mitteilungen dieser Station erscheinen dann im Fenster rechts (Rx Frequency).

Tipp 4: Drei Häkchen

Sobald Sie Ihre Sendefrequenz gewählt haben, möchten Sie nicht, dass diese auf die Frequenz der Station, die Sie zu kontaktieren versuchen, geändert wird. Sie können dies vermeiden, indem Sie das Kontrollkästchen *Hold Tx Freq.* aktivieren. Durch Markieren dieses Kästchens wird sichergestellt, dass die angegebene Sendefrequenz sich nicht automatisch ändert, wenn Sie auf dekodierten Text oder ein Signal im Wasserfall doppelklicken. Zwei weitere Kästchen, die Sie aktivieren sollten, sind *Auto Seq*, damit Ihre Station automatisch die richtige

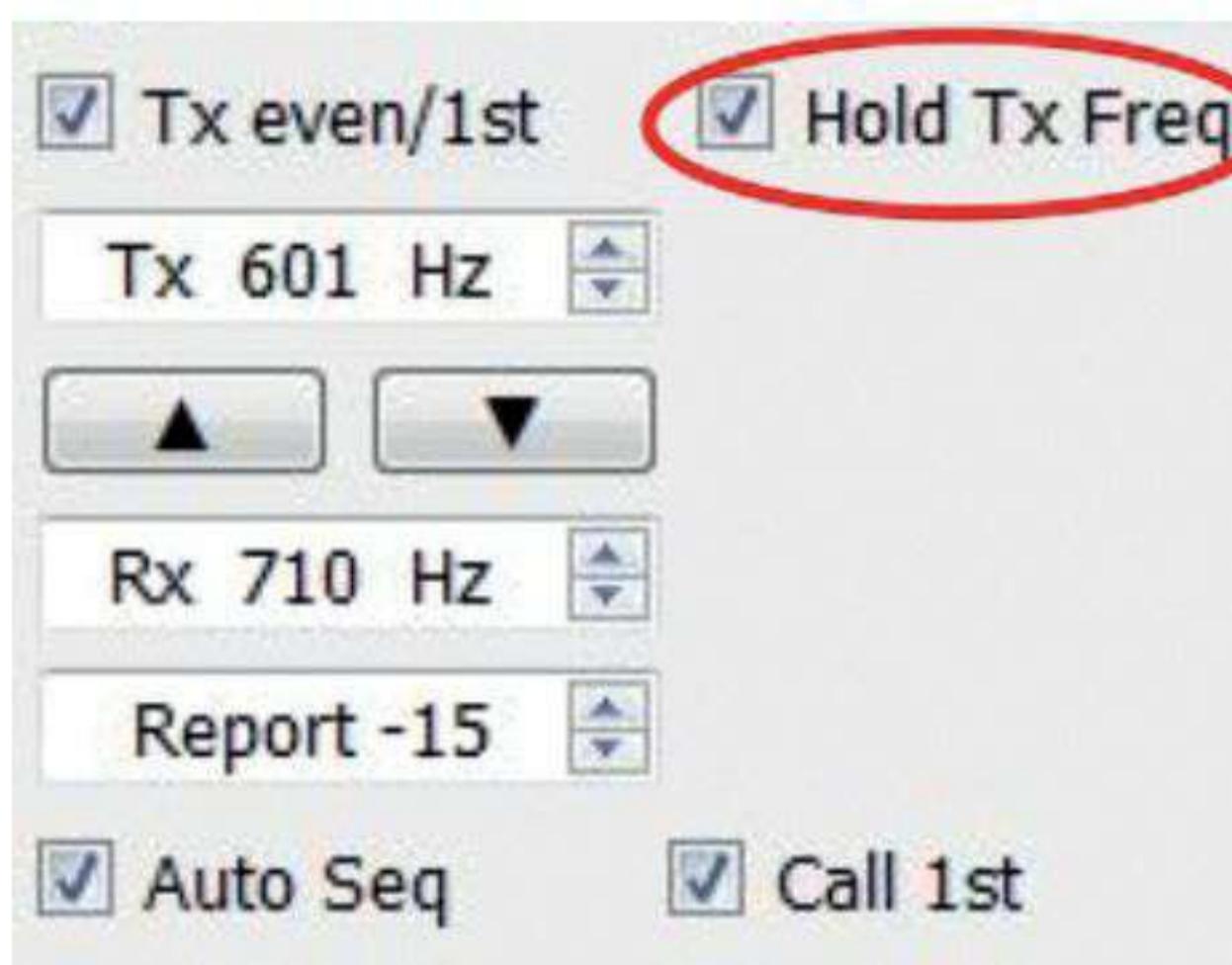


Bild 4: Drei Häkchen müssen gesetzt sein, wiTr hier zu sehen.

nächste Übertragung in der Reihenfolge sendet, und *Call 1st*, das bei einem CQ-Ruf ihrerseits eine automatische Antwort auf den ersten dekodierten Responder ermöglicht.

Tipp 5: Das richtige Band

Genau wie bei anderen Betriebsarten hängt die Wahl des zu verwendenden Amateurfunkbandes von vielen Faktoren ab, wie z.B. den aktuellen Ausbreitungsbedingungen, den vorhandenen Antennen und dem Zielgebiet der Aussendung. Weiterhin ist zu beachten, ob Sie in FT8 oder FT4 senden möchten: FT8-Aktivität findet man normalerweise auf allen Bändern, aber das neuere FT4-Protokoll findet man oft nur zur Zeit von Funkwettbewerben und auf dem 20- und 40-Meter-Band. Mit der Zeit wird die Zahl der FT4-

Nutzer sicherlich auf den anderen Bändern noch steigen.

Genau wie bei anderen Betriebsarten gibt es während der Contests vermehrte Aktivität. Auf einigen Bändern, wie z.B. 2 m, gibt es an den meisten Tagen nur selten FT8/FT4-Verkehr zu beobachten, aber an Tagen mit einem UKW-Contest ist viel los. In den letzten Jahren haben viele Wettbewerbe FT8/FT4 als QSO-Kontakte akzeptiert und es gibt sogar FT8/FT4-spezifische Wettbewerbe wie den World Wide Digi DX Contest.

Tipp 6: Die richtige ALC-Einstellung

ALC (Auto Level Control) ist die Technik, welche die Lautstärke der NF (Mikrofon oder FT8-Signal) bei leisen Passagen erhöht und an lauten Stellen vermindert. Man stellt die gewünschte - für FT8 eine geringe - Sendeleistung ein und stellt das SWR-Meter am TRX so ein, dass der ALC-Wert im S-Meter dargestellt wird. Das kann je nach Funkgerät ein horizontaler Balken oder ein Zeigerausschlag sein. Man stellt in WSJT-X den Schieberegler *PWR*, im WSJT-X-Fenster rechts zu sehen, langsam etwas herunter oder herauf, bis die ALC (am TRX) nicht mehr regelt. Hat die ALC nichts zu tun, ist der Pegel des NF-Signals der Soundkarte richtig justiert. In der FT8-Arbeitsanleitung heißt es dazu: „Obgleich FT8 ein FSK-Modus mit konstantem Träger ist, ist es sehr wichtig, eine Übersteuerung zu vermeiden, die das Signal verzerrt und verbreitert und Störsignale erzeugt. Stellen Sie die Sendepegel so ein, dass die gesamte Kette von

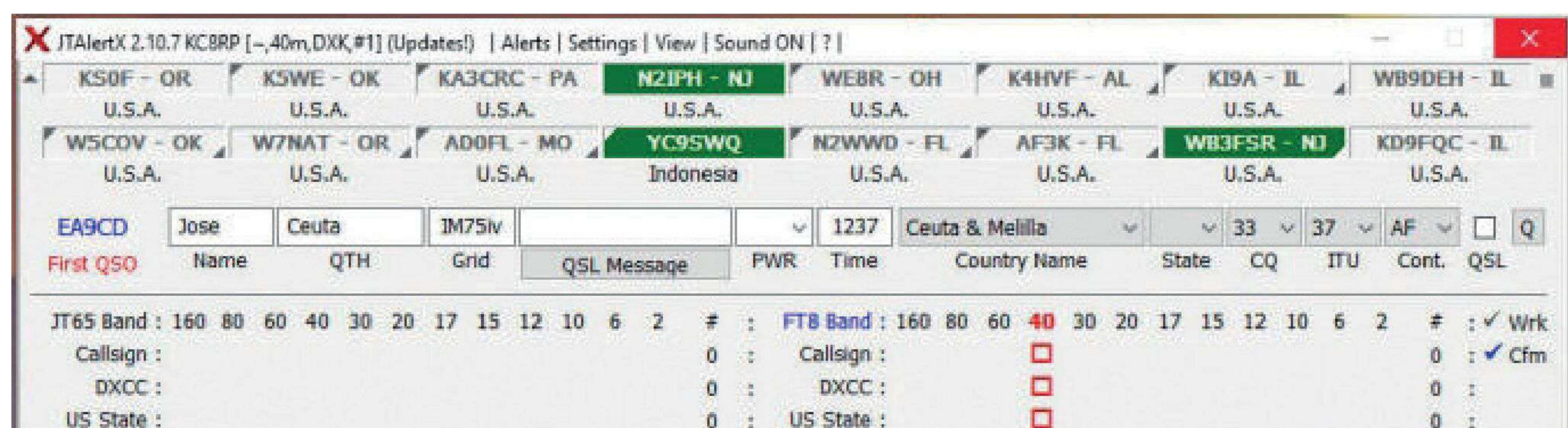


Bild 5: Das Fenster von JTAlert.

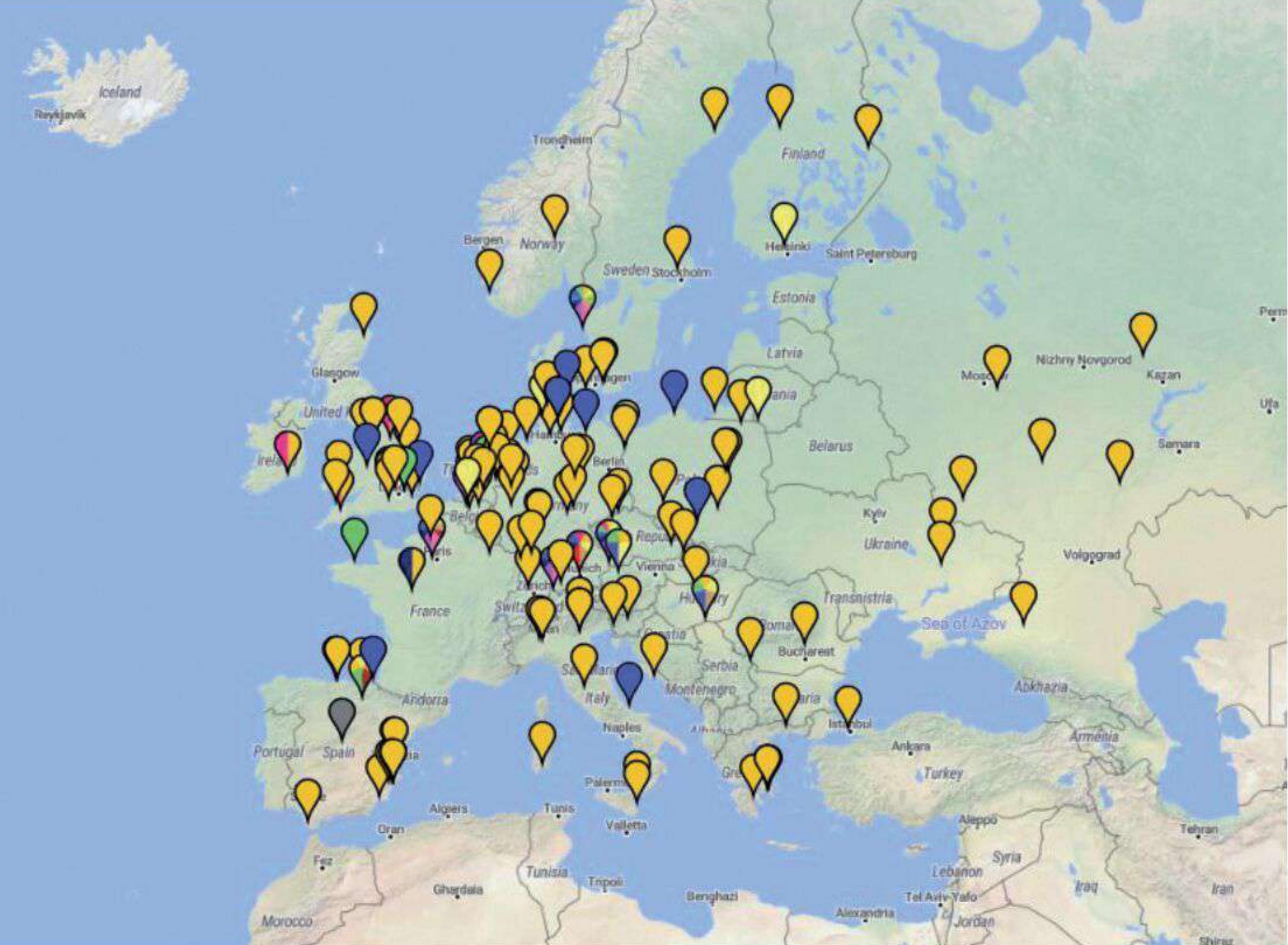


Bild 6: PSKReporter zeigt FT4-Kontakte der letzten 12 Stunden.

der NF-Erzeugung bis zur HF-Aussendung linear arbeitet.“ Kapitel 4 der Arbeitsanleitung widmet sich diesem Thema. Nachdem Sie die Einstellungen als gut erachtet, prüfen Sie die Breite ihrer Aussendung mittels eines WebSDR [7]. Im Wasserfalldiagramm vergleichen Sie ihre Bandbreite mit der Breite anderer Stationen.

Tipp 7: Mehr Komfort

Es gibt einige zusätzliche Programme und interaktive Websites, die Sie bei dem Betrieb mit FT8 und FT4 unterstützen.

JTAlert von Laurie Cowcher, VK3AMA, arbeitet als Hilfsprogramm mit WSJT-X und JTDX

zusammen. Die Software gleicht sich mit dem Logbuch ab und zeigt, welche Stationen bereits gearbeitet wurden und welche nicht. Bei Erkennen fehlender Rufzeichen, Landeskennung oder vieler anderer Merkmale – das ist in weiten Grenzen einstellbar – löst JTAlert einen Alarm aus und „weckt“ den Operator mit einem akustischen Signal. Die Software meldet sich auch, wenn das eigene Call gerufen wird. Sehr praktisch! JTAlert gibt es für den Mac nicht, aber eine Alternative: AlarmetJT gibt es für Mac und Linux und verfügt etwa über dieselbe Funktionalität wie JTAlert.

PSK Reporter: Ein Sammler für Empfangsberichte digitaler Stationen. Er löst die Frage: „Wer kann mich empfangen?“. Es zeigt auf einer Karte alle Stationen, die ihr zuvor gesendetes Signal empfangen konnten mit der Empfangs-Signalstärke. Die Webseite [4] informiert über die Ausbreitungsbedingungen in Echtzeit ohne aufwendig Kontakte zu YL/OMs in verschiedenen Kontinenten herstellen zu müssen oder Baken

zu hören. Pskreporter kann man auch nutzen, um die Ergebnisse vom Standort einer anderen Station aus zu betrachten.

HamSpots: Das ist ein weiterer Aggregator (Sammler) für digitale Amateurfunkspots [5]. Eine Anmeldung mit Rufzeichen und E-Mail-Adresse ist nötig, damit nur Rufzeicheninhaber Zutritt finden. Der Betreiber dieser Webseite verzichtet auf die Darstellung von Welt- oder Landkarten, sondern setzt auf die Anzeige aller Informationen in Form von Tabellen. Der Betreiber von Hamspot ist Autor von JTAlert. Spots lassen sich nach Bändern, mehreren Bändern oder Betriebsarten (auch FT8 und FT4) filtern (siehe Bild 7).

Tipp 8: DXpeditionen sind anders

Zum Arbeiten mit DXpeditionen und seltenen Stationen: Um das Gedränge (Pile-Up) zu vermeiden, das beim Auftreten seltener DX-Stationen auftritt, verwenden einige DX-Stationen und DXpeditionen zwei Techniken, um das Potenzial für QSOs zu maximieren:

DXpeditionen verwenden alternative Frequenzen außerhalb der in WSJT-X vordefinierten Frequenzen. Um sie zu arbeiten, müssen Sie die Frequenz in WSJT-X manuell eingeben, und zwar über dasselbe Dropdown-Auswahlfeld, das Sie normalerweise zum Wechseln des Bandes verwenden. Klicken Sie einfach in das Feld und geben Sie die gewünschte Frequenz über die Tastatur ein.

Und wie finden Sie die Frequenz? Genau wie bei anderen

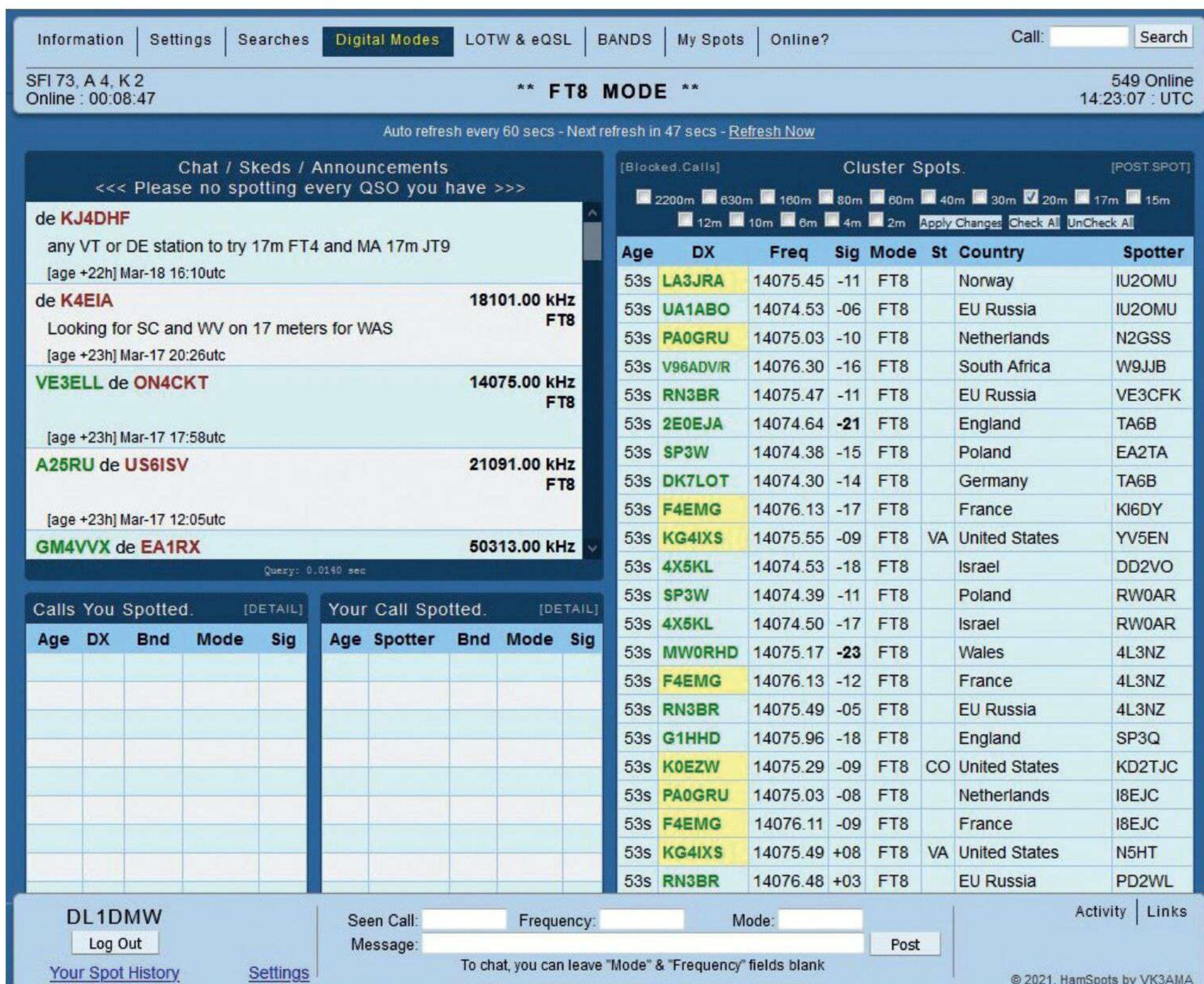


Bild 7: Die Tabelle zeigt FT8-Aktivitäten des 20-m-Bandes.

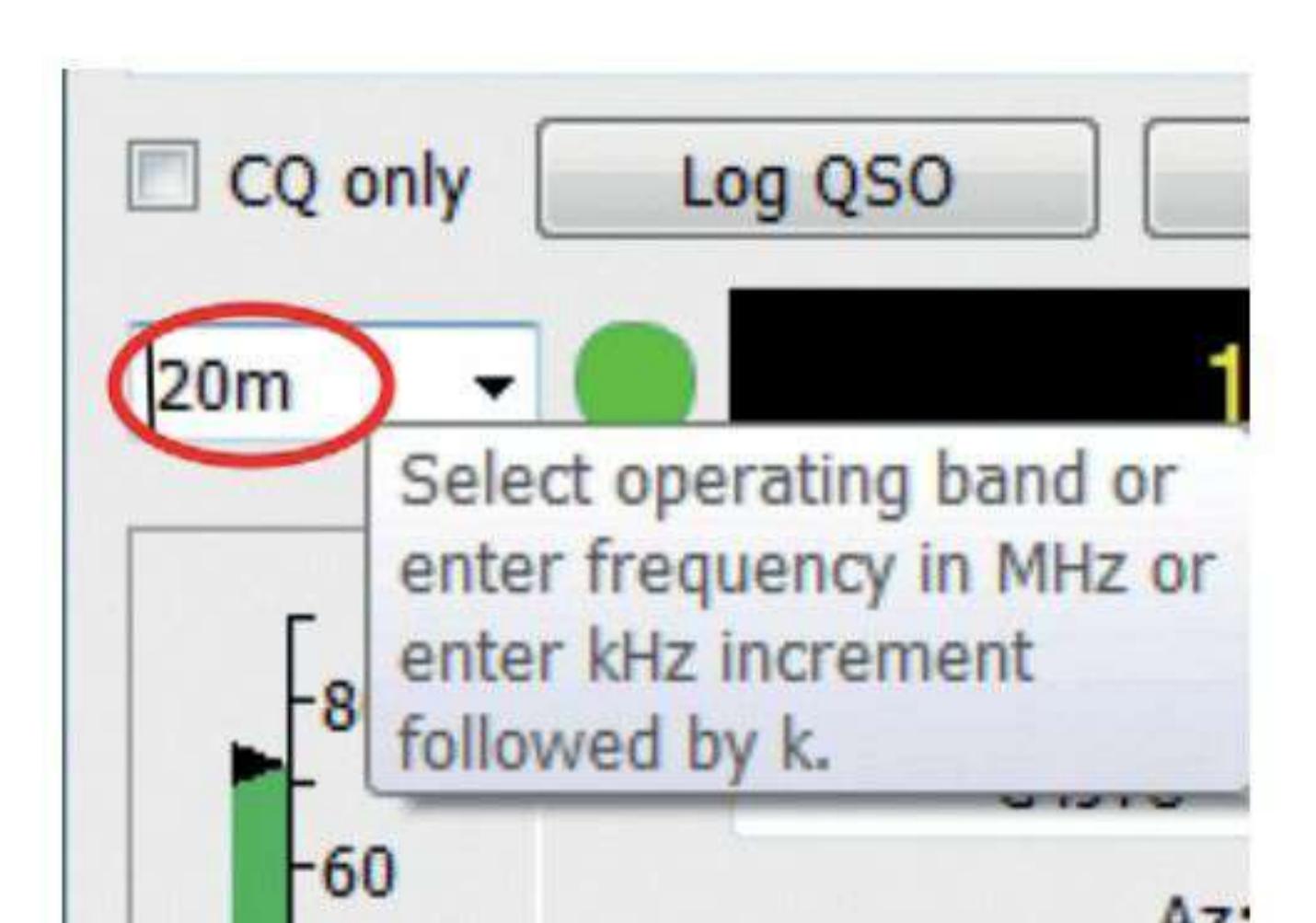


Bild 8: In dieses Feld kann man auch eine Frequenz im MHz eintragen (z.B. 14.105). Möchte man 100 kHz höher springen, schreibt man 100k in das Feld.

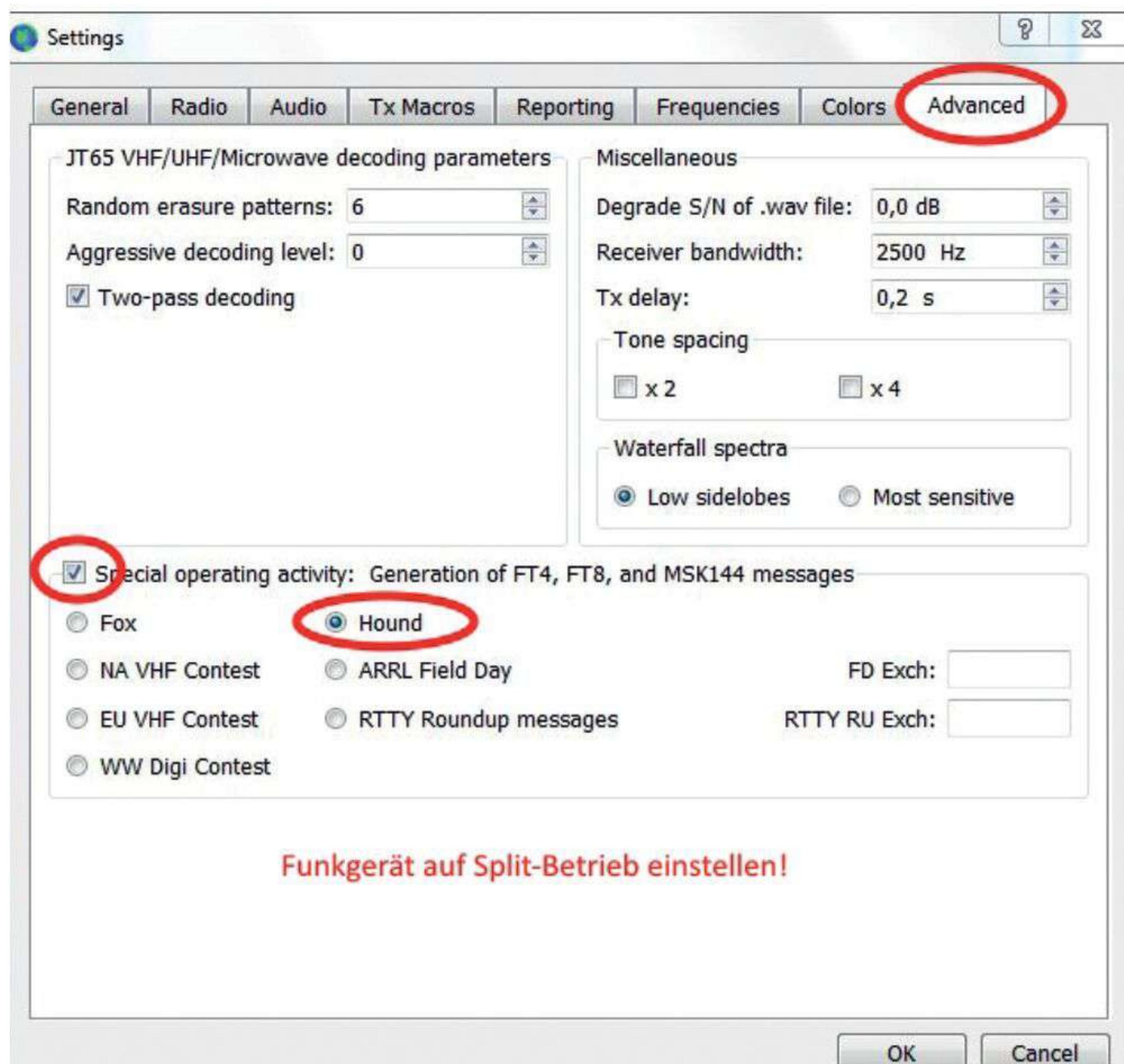


Bild 9: Für das Fox-Hound-Verfahren zum QSO mit einer DXpedition ist diese Auswahl zu treffen. Erfragen Sie auch die Sendefrequenz!

DX-Sendungen können Sie nach Spots bei PSK-Reporter oder Hamspot Ausschau halten oder einen Tipp von einem Freund erhalten. Auf der Webseite der DXpedition findet man evtl. Listen mit vorgeschlagenen Frequenzen.

Tipp 9:

Viele DXpeditionen verwenden aus Gründen der Effektivität und Zeitersparnis eine spezielle Version des FT8-Betriebs namens *Fox* und *Hound* (*Fuchs* und *Jäger*). Dabei ist die DXpedition der Fuchs und Sie der Jäger. Das Fox-Hound-Verfahren verändert die Arbeitsweise des QSO und muss in WSJT-X explizit eingeschaltet werden. Die Optionen *Fox/Hound* finden sich in dem *Settings* in der Registerkarte *Advanced*. Beachten Sie auch, dass DXpeditionen auf speziellen Frequenzen senden können.

Aktivieren Sie zuerst das Kontrollkästchen *Special operation activity* und wählen danach den spezifischen Typ, meist *Hound*. Fox und Hound teilt den ZF-Durchlassbereich für die Arbeit für DXpeditionen auf: Der Fox arbeitet auf dem unteren Teil des Durchlassbereichs, alle Hounds senden darüber. Zitat aus dem FT8-DXpeditions-Handbuch [6]: „Der Fuchs sendet bei Audiofre-

quenzen zwischen 300 und 900 Hz. Beim Senden von mehreren gleichzeitigen Signalen sind die Signale in Abständen von 60 Hz angeordnet. Jäger machen erste Anrufe überall im Bereich von 1000 bis 4000 Hz. Der Fuchs-Operator wird auf Jäger, die unter 1000 Hz rufen, nicht reagieren.“ Ein Fox kann auf mehrere Jäger gleichzeitig antworten. Die Auswahl *Fox* ist nur für die Station der DXpedition reserviert, also benutzen Sie diese Option nicht. Die weiteren Einstellungen aus Bild 9 verändern die Art und Weise, wie CQ gesendet wird, und das in Abhängigkeit vom jeweiligen Contest. Wählen Sie Ihren Contest aus. Lesen Sie auch die zum Fox-Hound-Verfahren existierende deutschsprachige Anleitung *FT8 DXpeditions-Handbuch!* Vergessen Sie nicht, das Häkchen bei *Special operation activity* zu entfernen, um zum normalen Betrieb zurückzukehren.

Tipp 10: Drei Mal Logbuch

Für die Arbeit mit WSJT-X gibt es drei Möglichkeiten der Führung eines Logbuchs: Version 1: Die WSJT-X führt das Logbuch und man importiert die ADIF-Datei manuell in das üblicherweise genutzte

Logbuchprogramm. Version 2 und drei sind praktischer: Voraussetzung ist hier, dass das Logbuchprogramm über einen UDP-Server und ein UDP-Port verfügt: Dann meldet WSJT-X über UDP jedes QSO an die Logbuch-Software - ein automatischer Import. Das ist beispielweise bei UCX-Log, Ham Radio Deluxe (HRD) und Log4OM möglich. Version 3: Sie benutzen JTAlert, das wiederum mit ALog, DXLab DX-Keeper, HRD, Log4OM und MixW zusammenarbeitet.

auf der korrekten Frequenz für das von Ihnen verwendete Band?

- Ist versehentlich ein Häkchen bei *Special operation activity* gesetzt und sie sind weder Fox noch Hound?
- Ist Ihr HF-Signal zu breit? Dann kann man Sie nicht dekodieren und stören andere Bandteilnehmer. Regeln Sie in WSJT-X den Regler PWR herunter und stellen Sie die ALC am TRX ein.
- Dekodieren Sie andere Stationen, aber es antwortet Ihnen niemand? Prüfen Sie auf PSK-Reporter, um zu sehen, ob andere Stationen Ihr Signal hören können.
- Funktioniert das Funkgerät in anderen Betriebsarten, z.B. CW, SSB?

Literatur/Verweise:

- [1] WSJT-X: <https://physics.princeton.edu/pulsar/k1jt/wsjtx.html>
- [2] Meinberg NTP: https://www.meinberg.de/german/sw/ntp.htm#ntp_stable
- [3] Dimension 4: <http://www.thinkman.com/dimension4/download.htm>
- [4] PSKReporter: <https://pskreporter.info/pskmap.html>
- [5] Hamspots: <https://hamspots.net/>
- [6] FT8-Doku: <https://physics.princeton.edu/pulsar/k1jt/wsjtx.html>
- [7] WebSDR: websdr.org

Michael Wöste, DL1DMW

FT8 Frequenzen, Anzeige im Display

Transceiver: USB
1,840 MHz
3,573 MHz
5,357 MHz
7,074 MHz
10,136 MHz
14,074 MHz
18,100 MHz
21,074 MHz
24,915 MHz
28,074 MHz
144,174 MHz

Liebe Leserinnen und Leser, die „Dummies“ sind nicht despektierlich gemeint, es soll ausdrücken, dass es sich hier um einen Beitrag für Einsteiger ohne Vorkenntnisse handelt und nicht um eine Abhandlung für Profis. Aus diesem Grund fängt es auch ganz harmlos an und Sie machen am besten mit,



SimSmith für Dummies:

Leistungsanpassung für Sender und Antenne

SimSmith finden Sie auf der virtuellen DVD: Betrachten Sie bitte Bild 2. Der Smith-Chart in Form eines Kreises besitzt in der Mitte einen 50-Ohm-Punkt. Fährt man auf der rot markierten X-Achse, der Linie der realen Widerstände, nach rechts, steigt der reelle Widerstand, geht man nach links, wird der Widerstand geringer. Klicken Sie im Smith-Chart mit der Maus genau auf diese horizontale Linie, informiert SimSmith über den Widerstandswert unten rechts neben dem Chart. Klicken Sie genau in der Mitte auf den Punkt, wird als Wert $Z=50$ zu lesen sein. Der Blindwiderstand ist hier Null. Wir wissen: Der reale Widerstand kann induktiv oder kapazitiv belastet sein. Fügen wir zwischen Sender und Antenne eine Kapazität (Kondensator) ein, wird sich der Blindanteil nach unten in den kapazitiven Bereich verschieben und bei Einfügen einer Induktivität (Spule) in den

SimSmith berechnet Anpassungsmaßnahmen zwischen Sender (Generator) und einer komplexen Last (Antenne). Die Darstellung der Real- und Blindwiderstände erfolgt mittels Smith Chart, die SWR-Anzeige im gewohnten X-Y-Koordinatensystem. Dieser Text richtet sich an HF-Dummies, die sich bisher gescheut haben, einen Smith Chart näher zu betrachten und um Berechnungen zur Anpassung einer Antenne einen großen Bogen machen.

induktiven Bereich nach oben. Wir fügen dem realen Widerstand also einen kapazitiven bzw. induktiven Blindwiderstand hinzu. Das werden wir später ausprobieren.

Betrachten wir das Schaltbild im Bild 3: Rechts ist der Generator (Sender) zu sehen und links als Z bezeichnet die Last bzw. die Antenne. Der Generator ist auf 50 Ohm eingestellt und auf eine Frequenz von 14 MHz. Die Antenne besitzt eine Anschlussimpedanz von 50 Ohm ohne Blindanteil, bezeichnet mit $Z_0=50$. Hier besteht

Leistungsanpassung – beide realen Widerstände sind gleich – die vollständige Leistung des Generators / Senders wird auf die Last / Antenne übertragen. Dem entsprechend zeigt der Smith-Chart einen Wert genau bei 50 Ohm ohne Blindanteil wie im Bild 2. Auf ihrem Bildschirm sollte es ebenso zu sehen sein, wenn Sie diese Schritte im Programm mitverfolgen. Verändern Sie nun den 50-Ohm-Wert der Antenne. Klicken Sie mit der Maus in das weiße 50-Ohm-Feld und betätigen das Mausrad. Betrachten Sie,

wie sich der Wert im Smith-Chart entlang der horizontalen, realen Widerstandslinie verändert. Stellen Sie danach den Wert auf exakt 50 Ohm zurück.

Mit Kondensator

Wir fügen dieser Schaltung einen Kondensator hinzu und sehen, was sich verändert. Ziehen Sie (mit der Maus) aus dem Bauteilepool unten links den vertikal gezeichneten Kondensator in die Mitte zwischen Sender und Antenne. Bild 4 zeigt, wie es

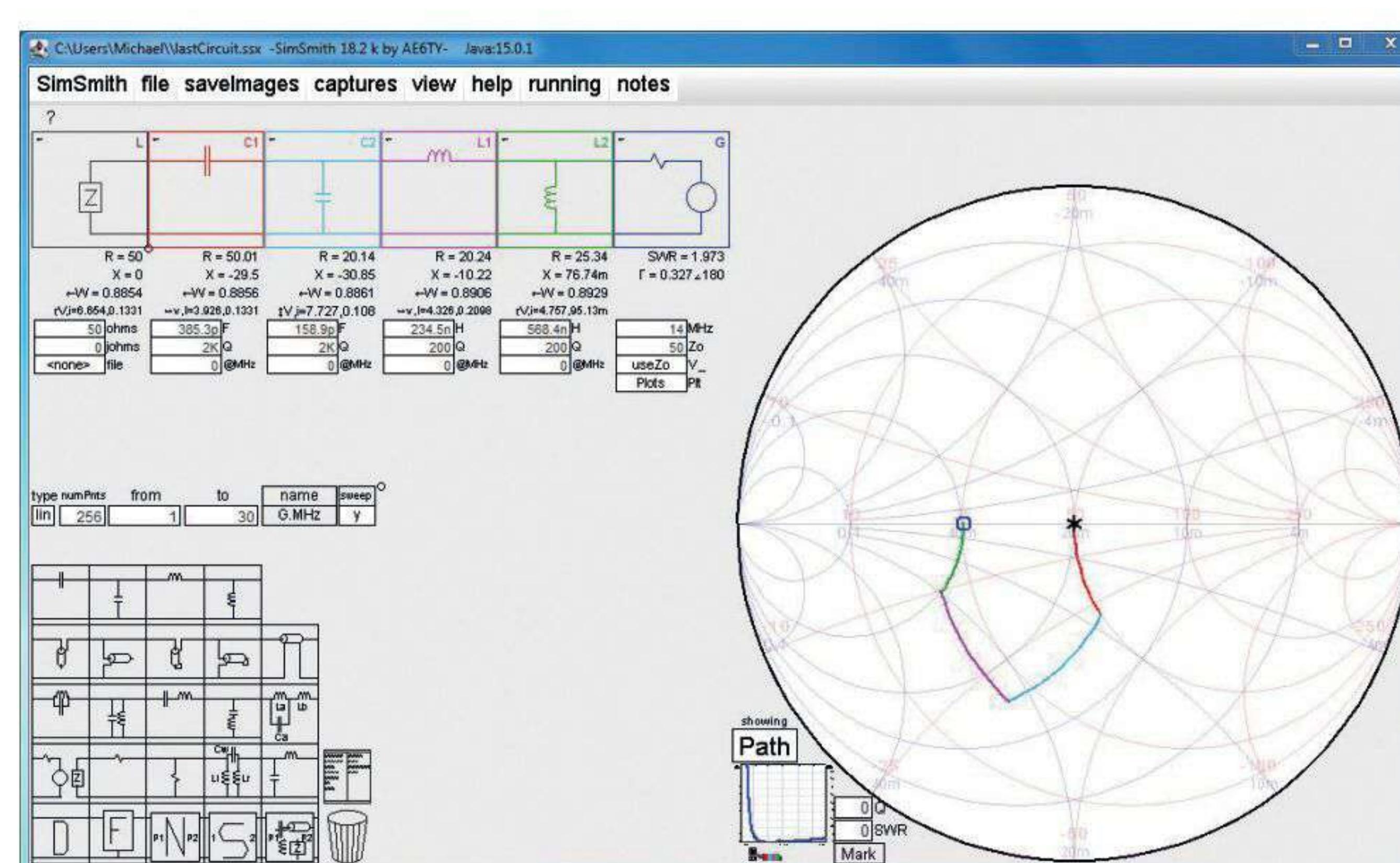


Bild 1: SimSmith unterstützt den Funkamateuer bei der Entwicklung von Anpassnetzwerken.

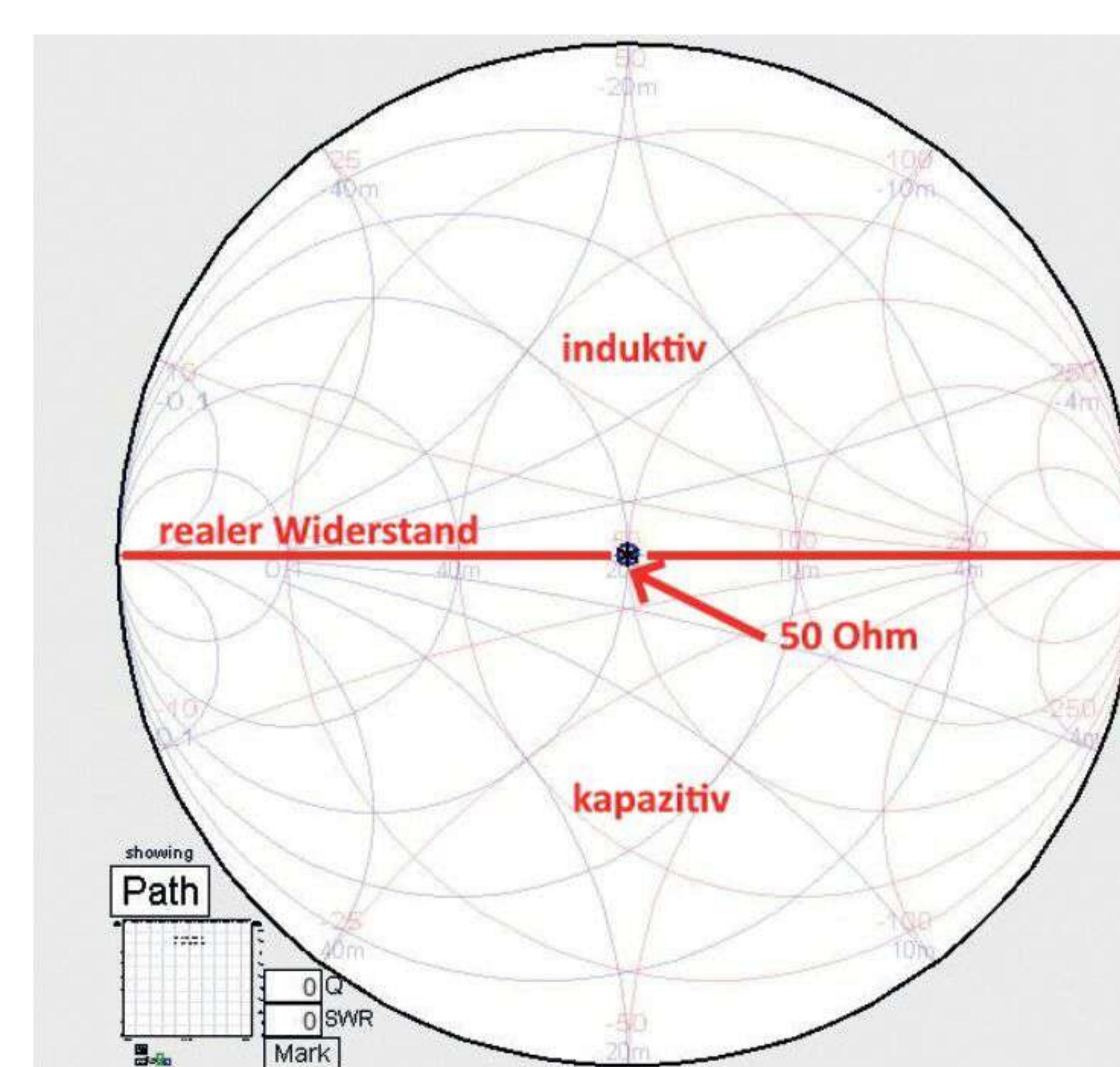


Bild 2: Das Smith-Diagramm ist so aufgebaut, dass sich der 50-Ohm-Punkt in der Mitte des Diagramms befindet.

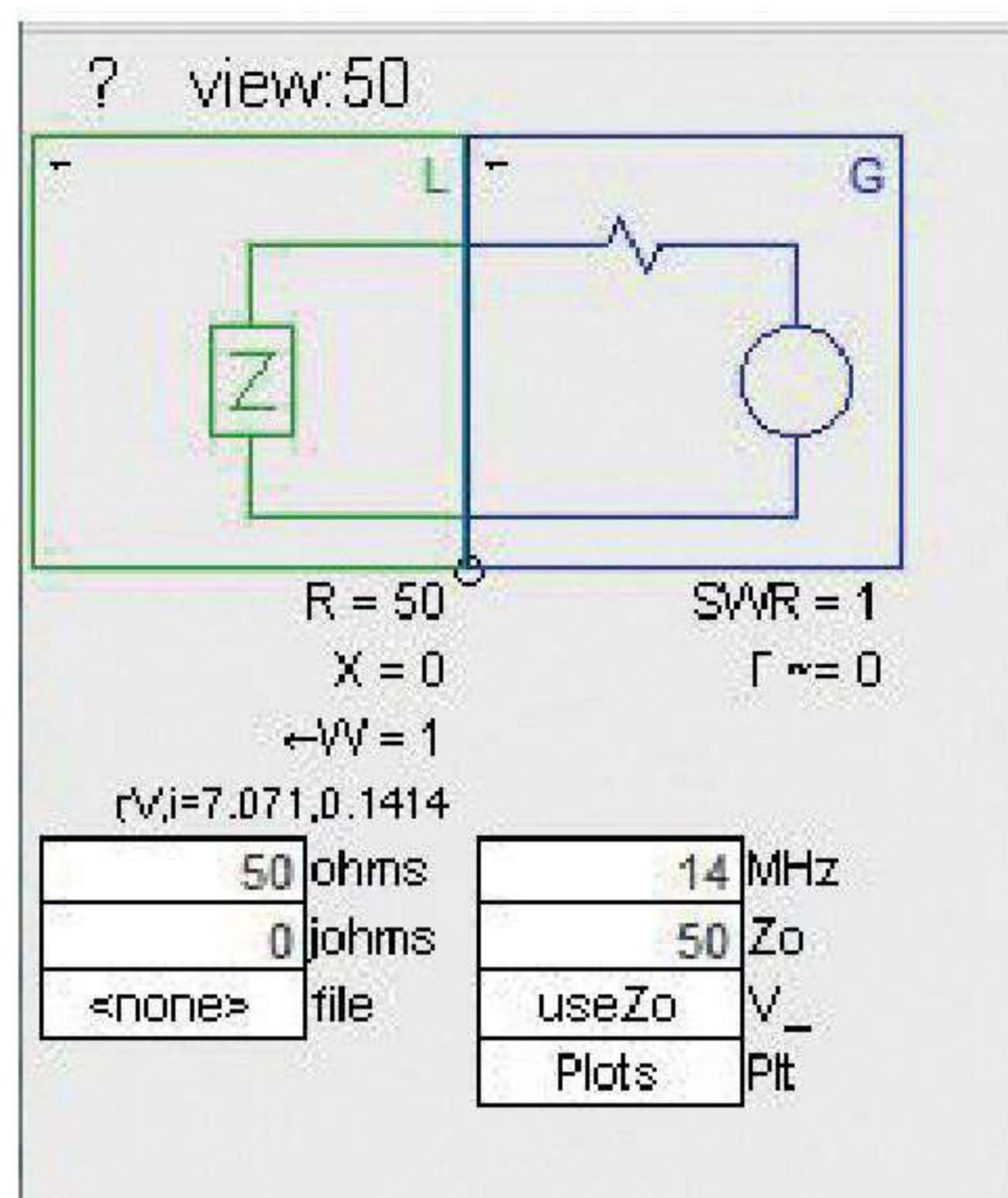


Bild 3: Sender links, Antenne rechts: Es ist Leistungsanpassung, beide Werte = 50 Ohm.

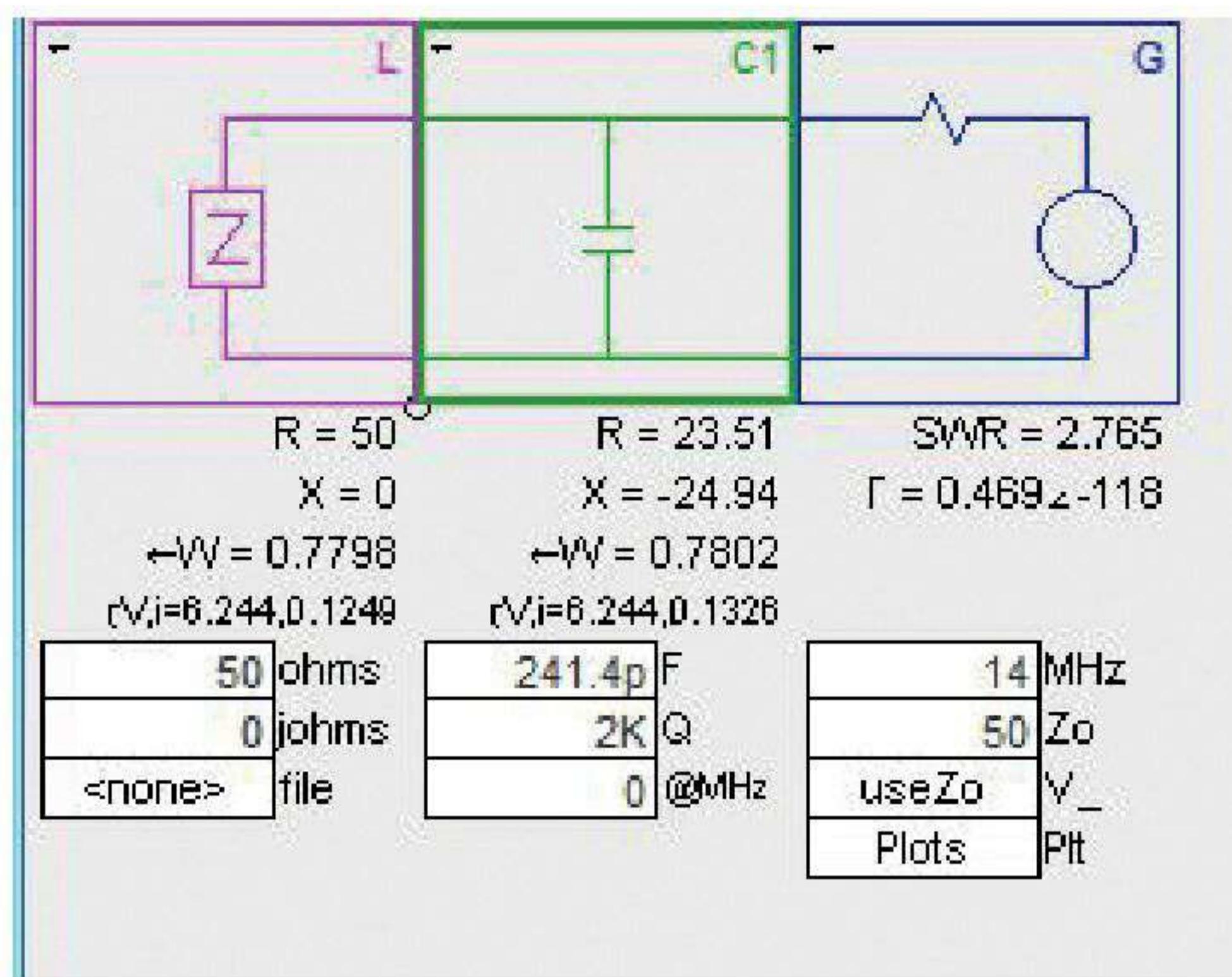


Bild 4: Ein Kondensator wurde in die Schaltung eingefügt. Klickt man auf den Wert des Kondensators, lässt sich der Wert mit dem Mausrad verändern und Veränderungen im Diagramm beobachten.

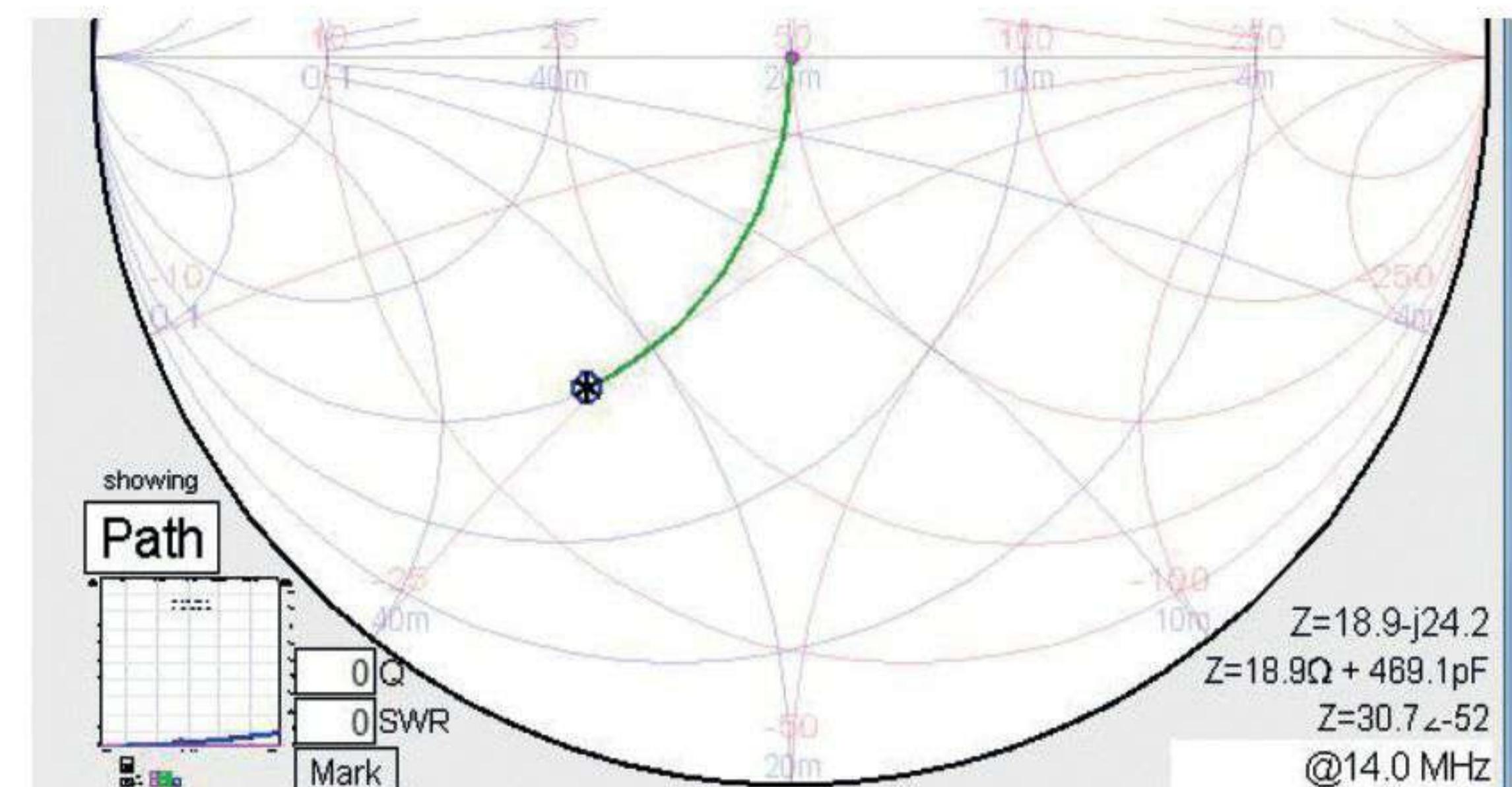


Bild 5: Der Smith-Chart beweist: Eine Kapazität wurde hinzugefügt. Der Kapazitive Blindanteil beträgt hier 24,2 Ohm.

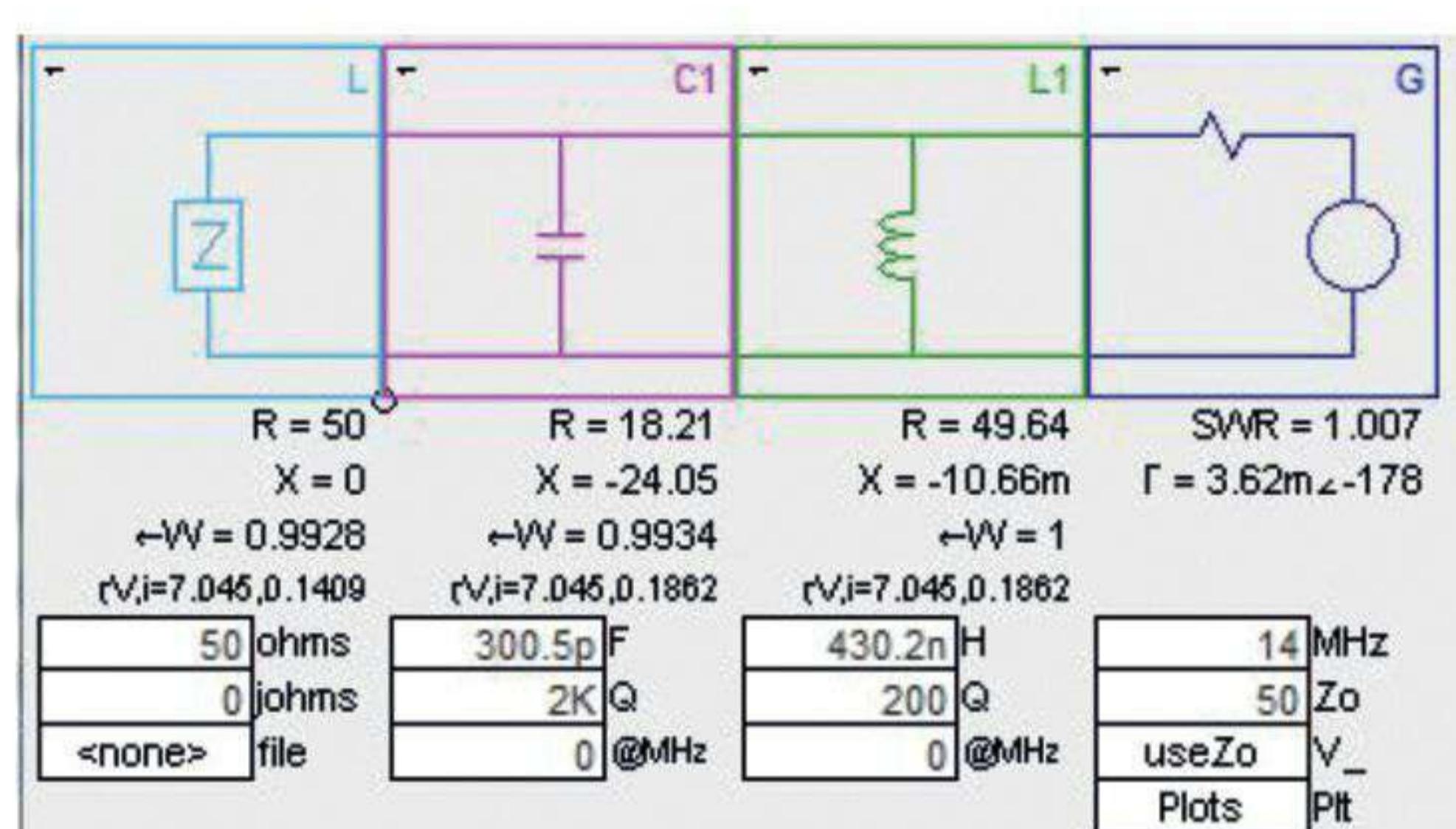


Bild 6: Nun kommt die Induktivität hinzu. Sie soll den durch den Kondensator verursachten kapazitiven Blindwiderstand kompensieren.

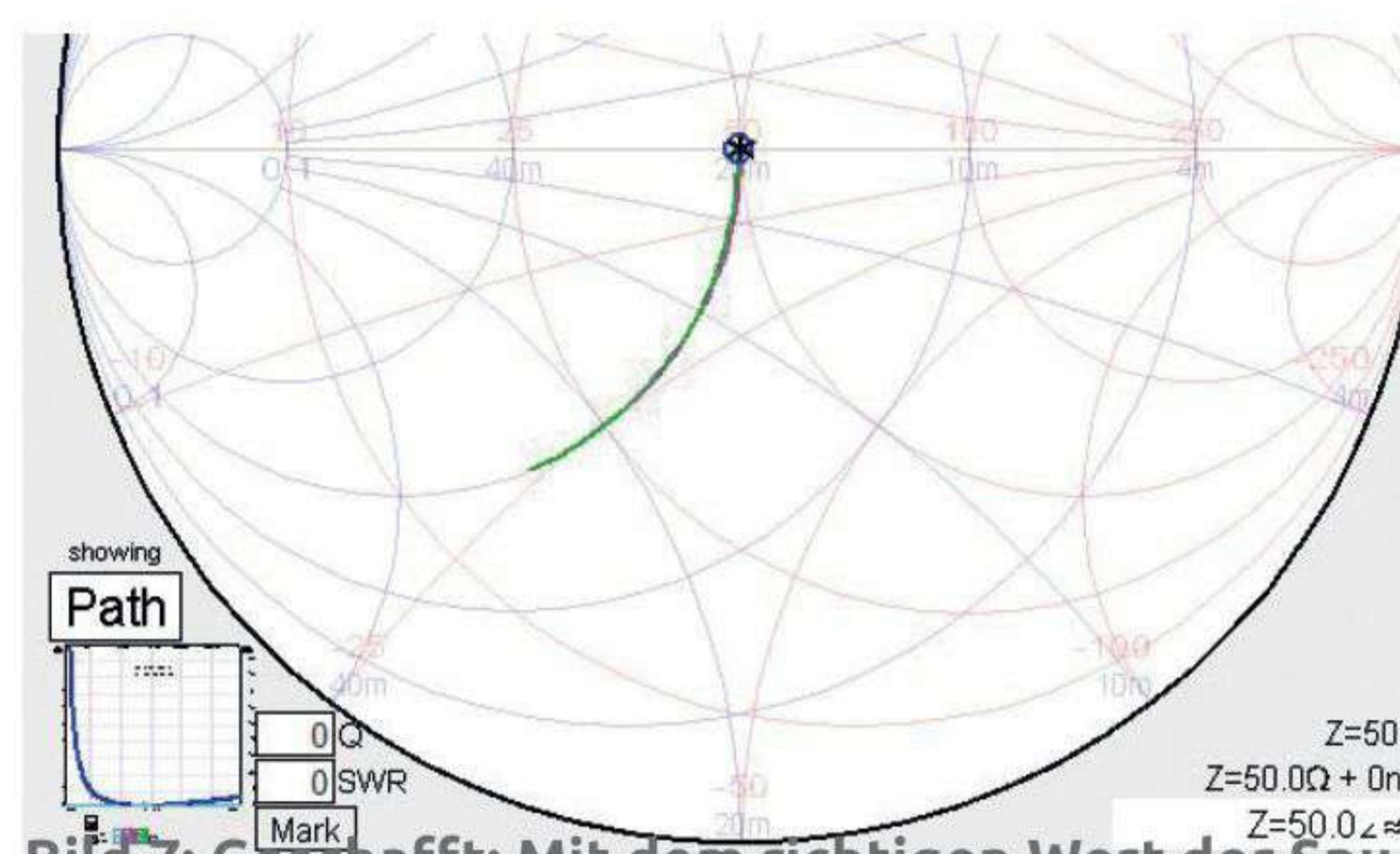


Bild 7: Geschafft: Mit dem richtigen Wert der Spule schafft es der reale Widerstand zurück auf den 50-Ohm-Punkt bei Null Ohm Blindanteil.

aussehen soll. Im Smith-Diagramm verrutscht der Wert folgerichtig in den kapazitiven Bereich. Verändern Sie mit dem Mausrad den Wert des Kondensators und sehen, wie sich der Wert mehr oder weniger in dem kapazitiven Bereich bewegt. Klicken Sie in den kleinen Kreis und setzen dort einen Marker in Form eines Sternchens, zeigt der Smith-Chart dem komplexen Widerstand unten rechts des Diagramm (Bild 5). Hier lese ich einen realen Widerstand von 18,9 Ohm und einen kapazitiven Blindwiderstand von 24,2 Ohm ab. Der kapazitive Blindanteil soll nun mit einem weiteren Bauelement kompensiert werden, um den Blindanteil zu kompensieren, also zu Null werden zu lassen.

Mit Spule

Was kann besser dazu geeignet sein, einen kapazitiven Blindanteil zu kompensieren als eine Induktivität? Ziehen Sie die vertikal gezeichnete Spule aus dem Bauelementepool zwischen Generator und Kondensator in die obige Schaltung. Bild 6 zeigt, wie es aussehen soll. Stellen Sie nun mit der Maus den Kondensatorwert auf ca. 300 pF ein und verändern anschließend – ebenfalls mit dem Mausrad – den Wert der Spule, bis der kapazitive Blindanteil zu Null wird, sich der kleine Kreis also wieder auf dem 50-Ohm-Punkt in der Mitte des Smith-Diagramms befindet (Bild 7). Es besteht wieder Leistungsanpassung: Die maximale Leistung des Senders

wird auf die Antenne übertragen, die kapazitive Belastung durch den Kondensator wurde durch eine Induktivität kompensiert. Beachten Sie bitte: Das gilt nur für die eingestellte Frequenz! Verändern Sie die Frequenz und sehen, was sich ändert.

Hatten Sie einen Aha-Effekt? So soll es sein! Wenn Sie möchten, testen Sie, ob sich ein in Serie eingefügter Kondensator durch eine in Serie eingefügte Induktivität ebenso kompensieren lässt wie die vertikalen Bauelemente zuvor (Bild 8). Bei richtiger Wahl der Bauteilwerte sollte das gelingen.

Die Kunst der Anpassung besteht prinzipiell darin, zu einer Abweichung in den kapazitiven oder induktiven Bereich geeignete Gegenmaßnahmen zu wählen, um sie zu kompensieren. Probieren Sie andere Bauelemente wie den Stub etc. aus und sehen, in welche Richtung sich die Blindanteile verändern (in Richtung kapazitiv/induktiv) und welche Gegenmaßnahmen möglich sind. Beispielsweise lässt sich ein in Serie eingefügter Stub durch eine seriell eingefügte Spule kompensieren. Finden Sie weitere Paare!

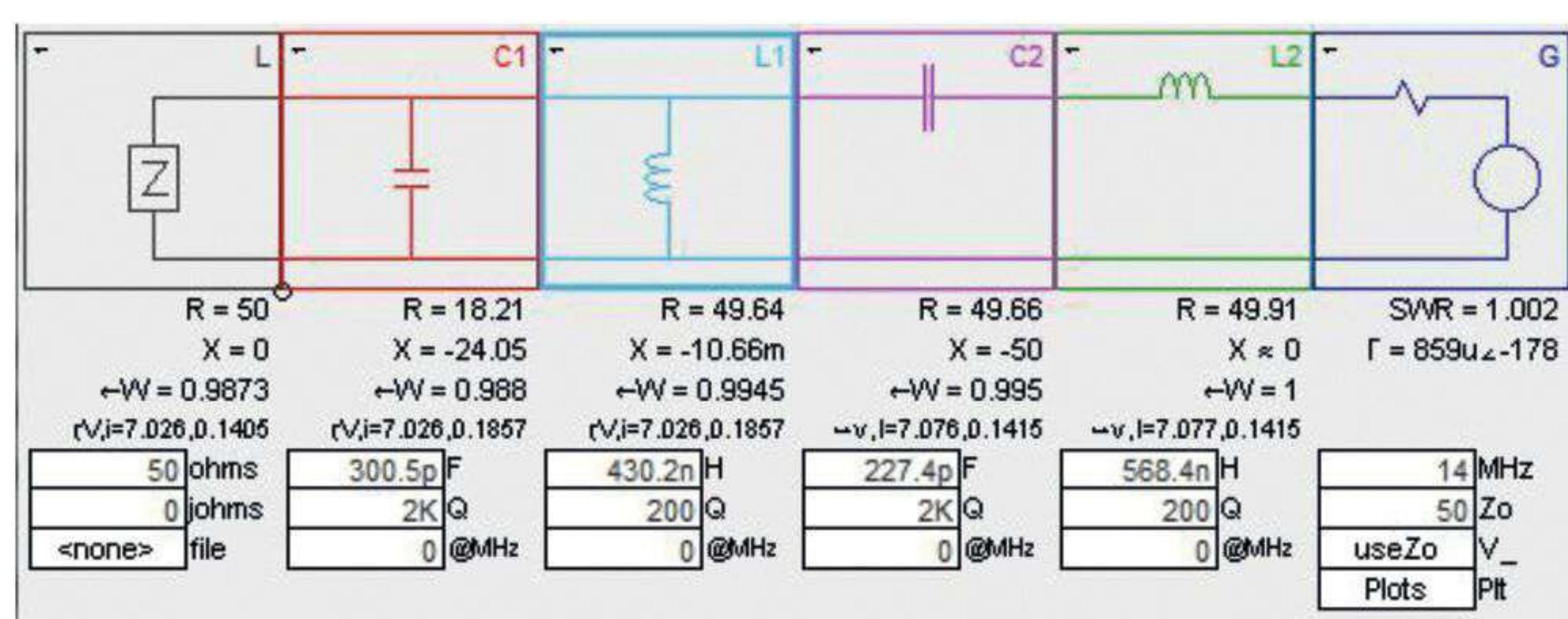


Bild 8: Doppelt kapazitiv belasten und kompensiert.

Weiteres Material

Das gezeigte Beispiel visualisiert den Begriff „Blindanteil“. Haben Sie Interesse, etwas mehr über die Fähigkeiten von SimSmith zu erfahren? Im Internet gibt es eine Menge Material dazu. Am besten eignen sich Videos, welche die Handhabung und Anwendung der Software sowie deren Funktionen vermutlich am anschaulichsten demonstrieren können. Auf Youtube.com gibt es vom Benutzer „LeoTV1968“ unter dem Titel „SimSmith Einführung – deutsch“ ein ca. 20 Minuten langes Video [1]. Etwa hundert Videos zum Thema Hochfrequenz unter anderen auch mit Anwendung von SimSmith hat Larry Benko, W0QE, in englischer Sprache gedreht. Eine Fundgrube für alle, die gut englisch verstehen. In seinen Videos gibt der pensionierte HF-Ingenieur sein umfangreiches Wissen an die interessierte Zuschauerschaft weiter. Volker Block hat sich in deutscher Sprache in seinem Video „Smith Diagramm für HF-Schrauber“ ohne Mathematik dem Thema Anpassung mit SimSmith genähert [4].

Literatur/Verweise:

- [1] SimSmith Einführung- deutsch: <https://www.youtube.com/watch?v=RUR6HojvA3o>
- [2] SimSmith-Videos, W0QE, engl: <http://www.w0qe.com/SimSmith.html>
- [3] SimSmith-Videos, WoQE, engl. auf Youtube: https://www.youtube.com/channel/UCKSyLSu4fm_1RH03Jvk4YQ/videos
- [4] SimSmith-Video von Volker Block: https://www.youtube.com/watch?v=lvb_hM1L5NM

Michael Wöste, DL1DMW

Bild 1: Zwei Brüder im Hörvergleich:
Tecsun PL-990x (rechts) vs. PL-880.



Ein Hörvergleich:

Als der Tecsun PL-880 vor ein paar Jahren in Fachzeitschriften rezensiert wurde, war mein Fazit: „Muss ich haben!“. Gesagt, getan. Als ich das hoch gelobte Gerät schließlich in meinen Händen hielt, waren die Erwartungen entsprechend hoch und, wie das oft so ist, wurden sie teilweise enttäuscht. Besonders bei SSB-Empfang konnte das Gerät – bei starken Signalen – nicht recht punkten: Die NF wirkte in den Spitzen verzerrt, während der FM- und AM-Betrieb keinerlei Anlass zu schlechten Bewertungen gab. Insgesamt habe ich mich mit dem



Tecsun PL-990x vs. PL-880

Zwei Tecsun-Weltempfänger im Hörvergleich: Der etwas ältere PL-880, damals in der Presse hoch gelobt, und der neuere, ein wenig größere Bruder PL-990x. Beide vertreten unterschiedliche Designperspektiven und sind deshalb nur bedingt vergleichbar. Dennoch: Ein subjektiver, direkter Hörvergleich wird zumindest aus der Sicht der Anwender neue Erkenntnisse hervorbringen.

Bild 2: Blick von rechts auf den PL-990x: Oben der Tuning-Knopf, darunter Fine-Tune, ganz unten die Lautstärkeinstellung.



alles in allem empfindlichen PL-880 „versöhnt“, doch zum SSB-Empfang habe ich ihn eher selten genutzt.

Und nun der Hörvergleich mit dem neueren PL-990x. Das ist kein um einen Si-Chip herum gebautes Gerät, sondern ein Dreifachsuper mit DSP-Demodulation. Der Neue ist etwas größer als der ältere Bruder, etwa 8 mm breiter, 5 mm höher und einige Millimeter tiefer. Bei beiden passt ein kraftvoller Akku vom Typ 18650 (3,7 Volt, 2600 mA) ins Akkufach und sorgt für eine lange Laufzeit. Insgesamt wirkt der PL-990x mit dunklen, quadratischen Knöpfen wertvoller als der PL-880. Gut verarbeitet sind beide. Vorteilhaft finde ich bei dem Neuen das matte Design im Gegensatz zum PL-880, dessen Gehäusefläche rund um LCD und Tasten glänzend ausgeführt wurde. Die Größe des Displays hat sich nicht verändert (65 x 27 mm)

und auch die Aufteilung des Displays ist sehr ähnlich. Beide zeigen oben rechts bei Empfang wahlweise die Uhrzeit oder die Signalspannung (in dB μ V) und das Signal-Rauschverhältnis (dB S/N) an. Die Wahl der AM-Bandbreite geschah beim PL-880 über eine einzige Taste unterhalb des Displays und rutschte beim PL-990x auf zwei Tasten genau über der ebenso gestalteten Shortwave-Bandauswahl (Wahl der Kurzwellenbänder). Die Lock-Taste zum Sperren unerwünschter Eingaben wanderte auf die Oberseite des PL-990x (beim PL-880 ist sie unterhalb der LCD angesiedelt). An der Zeitanzeige, die Funktion beider Timer, der Snooze und die Sleep-Funktion hat sich nichts Wesentliches verändert. Bewährtes darf auch bleiben. Gilt der externe Antenneneingang des PL-880 nur für die Kurzwelle, ist der mit *Antenna*

bezeichnete Klinkeneingang beim PL-990x für alle Bänder wirksam. Steckt man am PL-990x eine externe (Wurf-)Antenne an, darf man nicht vergessen, den Antennenumschalter von *Int* auf *Ext* zu schieben. Beide Empfänger verfügen über Line-Out und einen Kopfhörerausgang. Als Radiohörer mit der Vorerfahrung des Tecsun PL-880 bedient sich der Neue ohne einen Blick ins Handbuch auf Anhieb. Nach meiner persönlichen Ansicht ist die Bedienung noch etwas einfacher und eingängiger gestaltet als beim älteren Modell.

Audiovergleich

Der UKW-Empfang ist bei beiden Empfängern ausgewogen und sauber. Der PL-880 hört sich allgemein etwas mittenbetonter an, der Lautsprecher im PL-990x scheint etwas bassbetonter zu sein, unabhängig von

der Bandauswahl. Den Umschalter *Bass-Treble* gibt es bei beiden, hier jeweils in Stellung Treble. Der PL-990x empfängt die Bänder LW, MW und SW von 50 kHz durchgehend bis 29.999 kHz, UKW von 64 bis 108 MHz, wobei sich der UKW-Bereich an die regionalen Bandpläne aller Kontinente anpassen lässt (für Europa 87,5 bis 108 MHz). Der in Deutschland leider nur noch abends zu hörende Mittewellenbereich – wenn die D-Schicht nachlässt und Auslandssender hörbar werden – lässt sich zwischen den Rastern 9/10-kHz umschalten. Zur Erinnerung: Der PL-880 startete „erst“ bei 100 kHz und wies zwischen den Bändern LW und MW eine vernachlässigbare Empfangslücke von wenigen Kilohertz auf.

Dreht man beim PL-990x auf den Bändern LW und MW den VFO, hört man ein leises Klicken und das umso lauter, je niedriger die Empfangsfrequenz und umso ruhiger das Band ist. Auf KW ist davon nichts zu merken. Das fiel allerdings nur weitab von elektrischen Störquellen in der Natur auf, beim Betrieb im Haus wurde das leise Geräusch durch andere überlagert. Ein nettes Gimmick ist das sanfte Ausblenden der Lautstärke nach dem Ausschalten des PL-990x – mir gefällt das.

Insgesamt ist der PL990x gegenüber dem PL-880 auf den Bändern LW und MW hörbar empfindlicher als der PL-880: An einem milden Sommerabend habe ich ab 19.00 Uhr weitab von Störungen durch andere elektronische Geräte Empfangsversuche gestartet. Zuerst wurde die Langwelle eingestellt und auf 198 kHz wurde ein englischsprachiger Sender sehr leise gehört. Die Sportübertragung, vermutlich der BBC, war auf dem PL-880 zwar noch verständlich, aber mit hohem Rauschanteil vernehmbar. Auf dem PL-990x war die Sprache etwas lauter (bzw. das Signal-Rauschverhältnis war besser) und es war weniger verrauscht. Damit es möglichst gerecht zugeht, empfingen beide Geräte über die eingebaute Ferritantenne und waren exakt gleich ausgerichtet. Auch wurde – damit sich beide nicht gegenseitig beeinflussen – der jeweils nicht aktive Empfänger ausgeschaltet.



Bild 3: Blick auf die linke Schmalseite: Antenneneingang (alle Bänder), Ausgänge für Line-Out und Kopfhörer, USB-Buchse zum Laden des Akkus und für den Zugriff auf die SD-Karte.

Wenige Minuten später wurde ein leises, langsames „Morseignal“ auf 351 kHz gesichtet. Es handelte sich um einen unterbrochenen Pfeifton, empfangen in AM. Mit dem PL-880 war das Signal sehr leise aufzunehmen, gerade noch zu hören bei einem Rauschanteil von ca. 50 Prozent. Und der PL-990x? Deutlich hörbar, bei einem geringeren Rauschanteil von ca. 20 bis 25 Prozent. Zwei Pluspunkte für den PL-990x!

Nächster Empfang auf der Mittelwelle, 693 kHz: Ein sehr leises Signal eines AM-Senders, vermutlich in Osteuropa. Beim PL-880 geht das Signal im Rauschen unter. Der PL-990x dekodiert immerhin die Sprache als englisch, auch wenn die Worte nicht zu verstehen sind. Die Anzeige auf dem PL-990x: S/N von 0 dB bei 12 dB μ V.

Aber nicht immer gewinnt der PL-990x, manchmal ist die hellere, mittenbetonte NF des PL-880 ein akustischer Vorteil: Empfang in LSB auf dem 80-m-Band, 3610 kHz: Eine laute und eine sehr leise Station sind im Gespräch. Der PL-880 hört das relativ starke Signal mit guter Verständlichkeit, das Hintergrundrauschen ist deutlich hörbar, die Verständlichkeit ist etwas besser als beim PL-990. Jedoch: Das leise Signal der zweiten Station geht beim PL-880 im Rauschen unter. Nun der PL-990: Das laute Signal ist gut verständlich, aber mit mehr Bass im Rauschen, die leise Station bleibt auch im Fading gerade soeben lesbar. Anzeige beim PL-990: 12 dB μ V, S/N 00 dB. Hier gewinnen beide Empfänger: Der eine, weil er die NF etwas heller wiedergibt und der Neue, weil er empfindlicher ist.

Nehmen wir zwei starke Stationen auf 3684 kHz in LSB. Hier empfangen beide Empfänger gut und es ist akustisch kein Unterschied erkennbar. Das Rauschen ist bei beiden gering. Theorie: Wenn der „Input“ stimmt, verschwimmen die Unterschiede.

Wir wandern mit der Empfangsfrequenz auf 3955 kHz in AM. Ein Klavierkonzert. Im PL-880 ist es gut hörbar bei einem Rauschanteil von geschätzt 10 Prozent. Der PL-990x hört sich ebenso gut an, ist doch etwas voller und klarer, die Anzeige dort zeigt 28 dB μ V, S/N 10 dB. Das der PL-990x sich etwas klarer anhört, zeigen andere Empfangsversuche auf 3985 kHz und weitere.

Empfang einer Volmet-Station auf 5450 kHz in USB: Der PL-880 empfängt mit guten S/N, R5, Rauschen bei ca. 10%. Der PL-990x ebenso, jedoch bassbetonter, das hellere NF-Signal aus dem Lautsprecher des PL-880 ist aus dem Rauschen (in meinen Ohren) besser aufzunehmen. Es ist subjektiv. Inzwischen sind mehrere Stunden des Empfangs vergangen. Die Empfangsfrequenz ist nun 5860 kHz in AM und es ist ein schwacher BC-Sender im Lautsprecher hörbar. Der PL-880 meistert dies mit einem Rauschanteil von 50 Prozent, der PL-990x mit einem Rauschen, das bei geschätzt 20 Prozent deutlich geringer liegt.

Eine Viertelstunde später höre ich in das 40-m-Band hinein. Zwei OM unterhalten sich auf 7168 kHz, beide QTH Italien. Der Empfang ist bei beiden Empfängern gut (R5) bei wenig Rauschen. Auch hier ist kein Unterschied zu hören. Dies ist oft passiert: Die Unterschiede zeigen sich – wie beschrieben – besonders bei leisen Stationen – und hier ist der PL-990x im Vorteil. Wie gut der Empfang des Neuen sogar auf dem unteren LW-Bereich ist, verdeutlicht der USB-Empfang bei 77,5 kHz, dessen DCF77-Signal ich (in der Natur abseits von Störungen) deutlich lesen konnte! Es wurden mehrere verschiedene, auch höhere Frequenzen abgehört. Das Muster war jedoch immer dasselbe und brachte keine neuen Erkenntnisse. Nun ja, die oberen Bänder sind auch nicht immer offen, sodass ein Empfängervergleich z.B. auf dem 10-m-Band nur selten sinnvoll erscheint.



Bild 4: PL-990x von oben: Snooze-Funktion, Einstellungen zum LCD, Lock-Taster als Tastatursperre.

PL-990x als PC-Lautsprecher, SD-Abspieler...

Verbindet man die USB-Buchse, die auch zum Laden des Akkus im Gerät genutzt wird, über ein USB-Datenkabel mit dem PC, installiert (in diesem Fall) Windows einen neuen Gerätetreiber, der es ermöglicht, den PL-990x als PC-Lautsprecher zu nutzen. Mehr noch: Im Windows-Explorer ist nun ein neues Laufwerk verfügbar, das der SD-Karte des PL-990x zugeordnet ist, die man auf der Unterseite des Empfängers in den Schlitz einschieben und einrasten kann. Nun lassen sich vom PC aus Musikstücke, Hörspiele oder andere Audiodateien (in den Formaten WAV 16bit / 44.1kHz, FLAC / APE / WMA und MP3) auf die SD-Karte kopieren. So hat man „seine“ Musik immer dabei und spielt sie ab, wenn das Radioprogramm einmal nicht so prickelnd ist. Auch fungiert der PL-990x wahlweise als Bluetooth-Lautsprecher, der beispielsweise von einem Smartphone mit Musik beliefert wird. An die Möglichkeit des Mitschnitts einer Radiosendung auf die SD-Karte hat man leider nicht gedacht oder konnte dies technisch nicht realisieren, wie schade!

Im Ordner Artikel\PL-990x der virtuellen DVD gibt es die englische Anleitung als PDF sowie weitere Abbildungen.

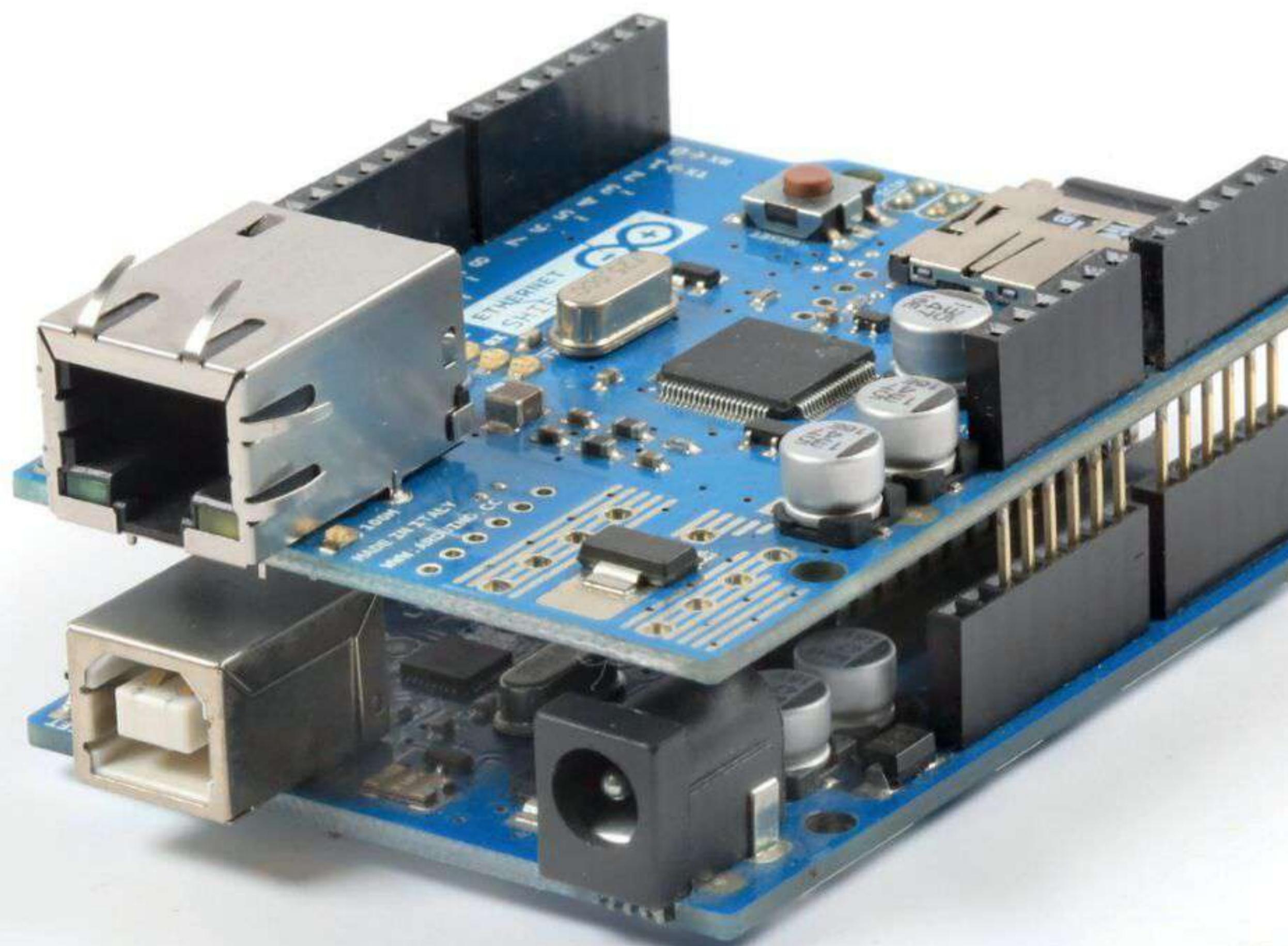
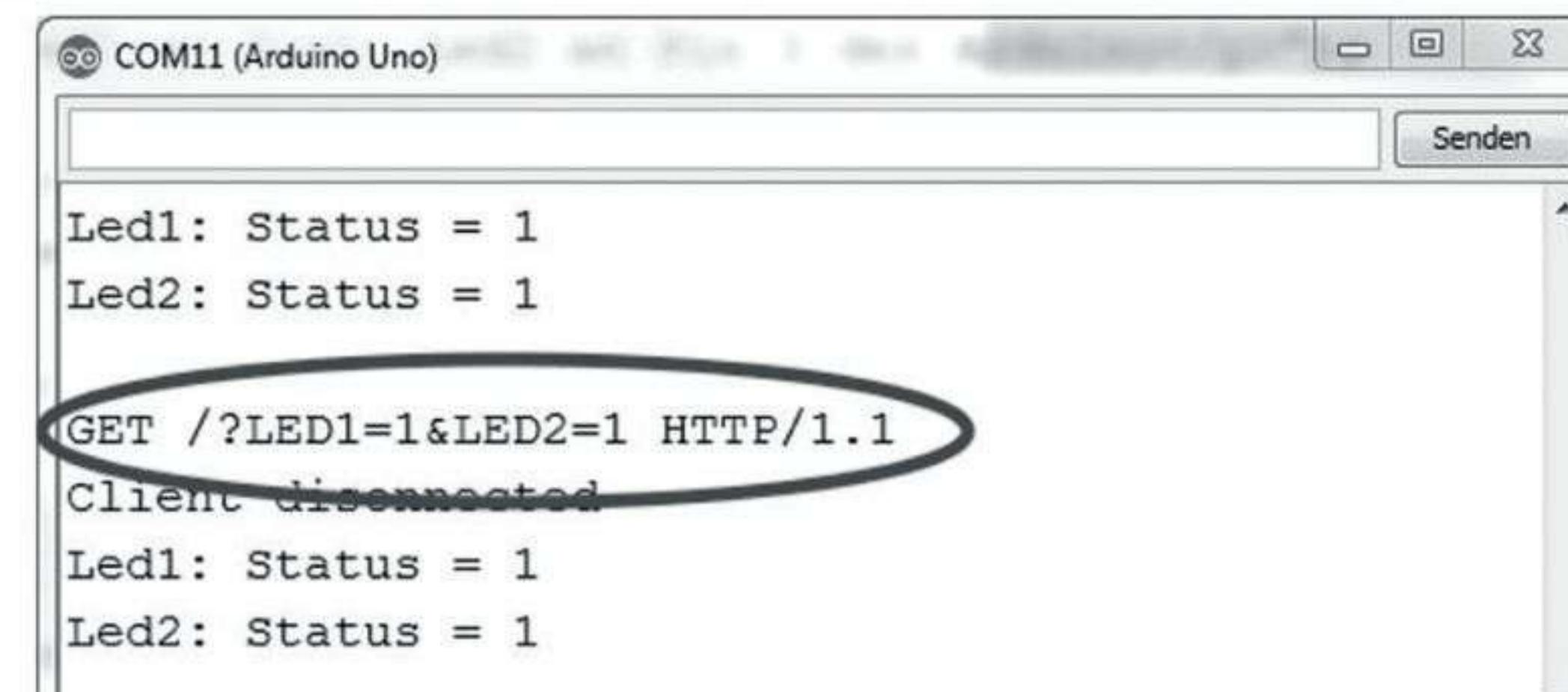
Literatur/Verweise:

- [1] Tecsun im engl. Wiki: <https://en.wikipedia.org/wiki/Tecsun>
- [2] Tecsun, Hersteller-Webseite: <http://www.tecsun-radios.com/>

Michael Wöste, DL1DMW

Vor etwa acht Jahren konnten Sie in dieser Heftreihe über einen Webserver lesen.

Wir haben einen Arduino UNO benutzt, um ihn über einen Browser zu steuern. Nun wollen wir einen Schritt weiter gehen, neue Methoden nutzen, um aus dem UNO schrittweise vom einfachsten Webserver hin zu einer ansprechenden Grafikdarstellung zu kommen. Wir schalten Aktoren und fragen digitale und analoge Werte ab. Das ist ein Beitrag zum Mitmachen! Wer mangels Hardware nicht mitmacht, wird dennoch etwas über die Art und Weise erfahren, wie Webserver und Webclient via HTTP interagieren.



Arduino Schalter-Status

Schalter oder Taster an Arduino Pin A0

Schalter: AUS

[Schalter-Status lesen](#)

```

<!DOCTYPE HTML>
<head>
<title>Arduino WebServer</title>
</head>
<html>
<body>
<meta http-equiv="refresh" content="5">
<h3>Arduino Webserver steuert zwei LED</h3>
<p>Dies ist eine feste Webseite des Webserver, <br>
die alle 5 Sekunden aufgefrischt wird<br>
Ledi an Pin2, Led2 an Pin 3 des Arduino</p>
<2> Status: LED1 ist AN // LED2 ist AN <br/>
<3> <FORM action = "http://192.168.1.31/"><input type = "radio" name = "LED1" value = "1">Ledi AN <br><input type = "radio" name = "LED1" value = "0">Ledi AUS </p><input type = "radio" name = "LED2" value = "1">Led2 AN <br><input type = "radio" name = "LED2" value = "0">Led2 AUS </p><input type = "submit" value = "Absenden"> </FORM> </p></body>
</html>

```

Arduino Webserver steuert zwei LED

Dies ist eine feste Webseite des Webserver, die alle 5 Sekunden aufgefrischt wird. Ledi an Pin2, Led2 an Pin 3 des Arduino

Status: LED1 ist AN // LED2 ist AN

Led1 AN

Led1 AUS

Led2 AN

Led2 AUS

[Absenden](#)

Bild 1: Webserver auf einem Mikrocontroller können unterschiedlich komplex ausfallen. Manchmal gelangt man schnell an die Speichergrenzen des Controllers.

Arduino UNO und Ethernet Shield:

Schalten und Walten per LAN



Webserver werden verwendet, um Informationen bereit zu stellen, auf die von einem Webbrower aus der Ferne zugegriffen werden kann. Hier läuft der Webbrower – z. B. der Mozilla Firefox – auf einem beliebigen Computer, der im selben Netzwerk (LAN) wie der Arduino angeschlossen ist. Sie sollten Ihren Arduino über ein Kabel an den Router oder einen LAN-Switch anschließen, ebenso wie dies beim PC der Fall ist. Mit einem Browser auf dem PC und dem Aufruf der Webseite des Arduino Webservers ist es möglich z. B. das Ein- und Ausschalten einer LED zu veranlassen. Auch der umgekehrte Weg der

Informationen ist möglich: Das Auslesen des Zustands eines Schalters, die Stellung eines Potentiometers oder Werte eines Sensors können erfragt und auf einer Webseite zur Anzeige gebracht werden. Der Beitrag vermittelt, was zum Aufbau eines Webservers erforderlich ist, einschließlich eines Überblicks über die Technologien wie HTTP, HTML, CSS, JavaScript und AJAX. Es startet mit den Grundlagen des HTTP einer einfachen, fixen Webseite auf dem Arduino und baut auf diesen schrittweise auf.

Wir benutzen ein Arduino Uno oder eine dazu kompatible Platine und – zum Zugang zum LAN – das

Arduino Ethernet Shield oder ein baugleiches Modell. Ein Ethernet-Kabel verbindet den Arduino mit einem Hub oder direkt mit dem Netzwerk-Router. Ein USB-Kabel sorgt für die Stromversorgung und Programmierung des Arduino UNO. Ein SD-Kartenslot befindet sich auf dem Ethernet Shield und speichert im Folgenden die Webseite.

Der simpelste Arduino-Webserver

Wir starten mit einem sehr einfachen Webserver, der eine einzelne Webseite unter Verwendung des Arduino-Ethernet-Shields dem LAN zur Verfügung stellt. Man kann sie dann mit einem Internet-

Browser aufrufen. Die SD-Karte wird in diesem ersten Beispiel noch nicht verwendet, da der Inhalt der Webseite fest im Arduino-Sketch (Programm) enthalten ist. Das Listing hat die Bezeichnung WebserverL1.ino und Sie finden es auf der Virtuellen DVD im Ordner Artikel\Webservice\ArduinoSketches. Installieren Sie die Arduino IDE auf Ihren PC, falls noch nicht geschehen und kopieren alle Ordner WebserverL1 und weitere in das Arduino Projektverzeichnis, dass sich in der Regel auf LW:\Benutzer\ Dokumente\Arduino befindet. Starten Sie die Arduino IDE und laden den Sketch WebserverL1.ino. Sie sehen folgende Zeilen, die Sie auf Ihre lokalen Gegebenheiten des LAN anpassen:

Arduino Analoganzeige

Analog Poti an A0: 403



```
// Ändern Sie hier die MAC-Adresse auf die Adresse Ihres Ethernet-Shields
byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x9D, 0xB5 };
```

```
// Passen Sie die IP-Adresse auf die Gegebenheiten Ihres LAN an.
IPAddress ip(192,168,1,31);
```

Ändern Sie die MAC-Adresse so, dass sie mit dem Aufkleber auf der Unterseite Ihres Ethernet Shields übereinstimmt. Ändern Sie auch die IP-Adresse, sodass sie in den IP-Adressbereich Ihres Netzwerks passt. Welcher Adressraum gültig ist, kann man unter Windows mit dem Befehl `ipconfig -all` in Erfahrung bringen. Hier reicht der Adressraum des LAN von 192.168.1.1 bis 192.168.1.255. Dem Arduino wird in diesem Fall die IP-Adresse 192.168.1.31 zugewiesen, weil sie bisher nicht durch andere Geräte belegt ist.

Schließen Sie nun den Arduino UNO über ein USB-Kabel an den PC an und das Ethernet Shield mittels des Ethernet-Kabels an den Router oder einen LAN-Switch. Kompilieren Sie den Sketch und laden ihn auf den Arduino. Anschließend öffnen Sie einen Webbrowser – ich verwende gern den Mozilla Firefox – auf einem PC, der sich im selben Netzwerk wie der Arduino befindet. Surfen Sie zum Arduino, indem Sie die IP-Adresse des Arduino in das URL-Feld des Browsers eingeben, z. B. hier lokal die 192.168.1.31. Geht alles gut, zeigt der Browser die Webseite wie in Bild 2 zu sehen ist.

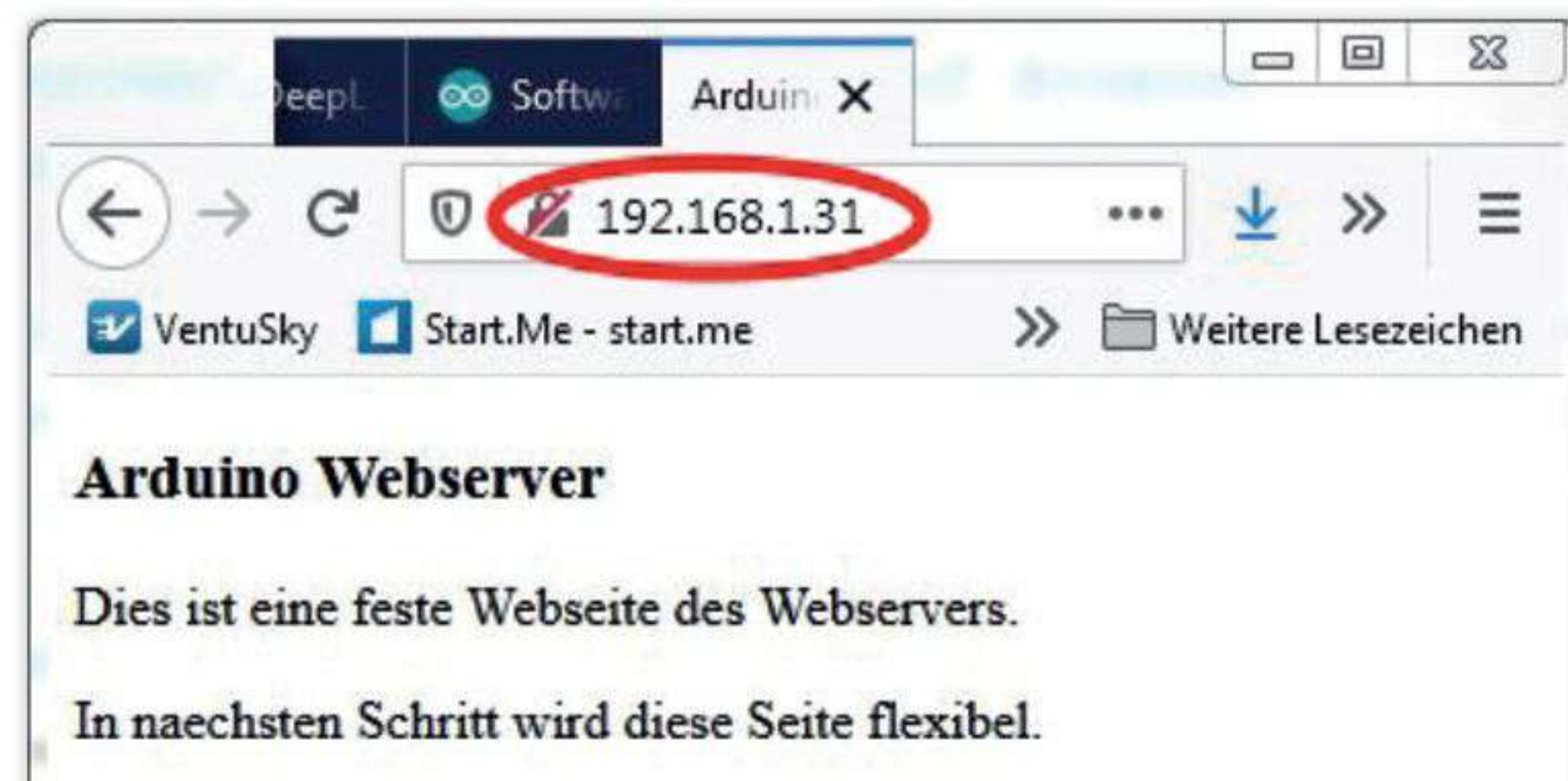


Bild 2: Der Firefox Browser zeigt die erste Webseite des Arduino Web-servers an.

Falls etwas schief geht: Sollten Sie keine Verbindung zum Arduino UNO herstellen können, versuchen Sie es mit einem Reset durch Betätigen der Reset-Taste auf dem Ethernet Shield und geben danach erneut die IP-Adresse in die Adresszeile des Webbrowser ein (in Bild 2 rot markiert). Stellen Sie sicher, dass Sie die richtige IP-Adresse des

Arduino für den Adressbereich Ihres Netzwerks eingestellt haben. Die ersten drei Zahlen der IP-Adresse (z.B. 192.168.1) müssen zu Ihrem Netzwerk passen (ebenfalls 192.168.1 oder bei Fritz-Boxen auch oft 192.168.178). Die letzte der vier Nummer muss eindeutig sein - d. h. es muss das einzige Gerät im Netzwerk mit dieser Nummer sein.

Gateway und Subnetzmaske: Versuchen Sie, das Netzwerk-Gateway und die Subnetzmaske in den Sketch anzugeben (diese Zeilen sind im Sketch zurzeit kommentiert), wenn es immer noch Probleme mit der Netzwerkverbindung gibt. Sie müssen die Adressen im folgenden Code ändern, damit sie zu Ihrem Netzwerk passen. Fügen Sie das Gateway und das Subnetz unter der MAC-Adresse in der Skizze hinzu, im Sketch reicht es in den Zeilen das Kommentarzeichen (//) zu entfernen und die Werte anzupassen:

```
byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x9D, 0xB5 };
IPAddress ip(192,168,1,31);
IPAddress dns1(192,168,1,1);
IPAddress gateway(192,168,1,1);
IPAddress subnet(255,255,255,0);
```

IP-Adresse des Arduino-Servers surfen, sendet der Webbrowser (auch Client genannt) eine Anfrage an den Webserver wie folgt:

```
GET /HTTP/1.1\r\n
Host: 192.168.1.31\r\n
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:87.0) Gecko/20100101
Firefox/87.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
webp,*/*;q=0.8\r\n
Accept-Language: de,en-US;q=0.7,en;q=0.3\r\n
Accept-Encoding: gzip, deflate\r\n
DNT: 1\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
\r\n
```

Zeichen automatisch an das Ende jeder Zeile an. Die `println()`-Funktion am Ende der HTTP-Antwort sendet lediglich das `\r\n` ohne einen

Die Informationen in der Anfrage unterscheiden sich je nach Browser und Betriebssystem, von dem die Anfrage gesendet wird. Hier ist es der Firefox-Browser unter Windows. Die `\r\n`-Zeichen, die Sie am Ende jeder Textzeile sehen, sind nicht sichtbare Zeichen (nicht druckbare Zeichen). `\r` ist das Wagenrücklaufzeichen und `\n` ist das Zeilenvorschubzeichen (oder Zeilenumbruchzeichen). Beide zusammen ergeben einen Zeilenwechsel. Die

letzte Zeile der Anforderung lautet `\r\n` ohne vorangestellten Text. Das ist eine Leerzeile, auf die der Arduino-Sketch prüft, bevor er eine Antwort an

den Client-Webbrowser sendet. Schauen Sie sich in der Arduino IDE den Quellcode des Sketches an. Er liest jedes Zeichen aus der obigen Anforderung und weiß, wann das Ende erreicht ist, da er die Leerzeile findet – ohne den zuvor empfangenen Text auszuwerten. Was soll der Server auch tun, als die einzige vorhandene Webseite zu senden?

Nachdem der Server die Anfrage für eine Webseite vom Client erhalten hat, sendet er zuerst eine Standard-HTTP-Antwort und danach schließt sich die Webseite selbst an. Die vom Arduino gesendete Antwort lautet wie nebenstehend.►

In der obigen Antwort werden wieder die nicht sichtbaren Zeichen `\r\n` angezeigt. Die `println("...")`-Funktion im Sketch fügt die `\r\n`-

Text. Anfrage und Antwort sind Teil des HTTP (Hypertext Transfer Protokolls).

Nachdem der Server die HTTP-Antwort gesendet hat, übermittelt er die eigentliche Webseite, die dann im Browser angezeigt wird. Sie besteht aus Text mit HTML-Tags (ein Tag ist so etwas wie eine Markierung). Sie sehen die Tags im Browser nicht, da sie vom Browser interpretiert werden und ihm mitteilen, dass es sich um das Protokoll HTML handelt (<http>), die Seite einen Titel besitzt (<title>), einen Informationsteil (<body>) und den darin enthaltenen Text (<h3> für die Überschrift 3. Grades und <p> für einen Absatz). Um den HTML-Quellcode zu betrachten, klicken Sie im Browser mit der rechten Maustaste auf die Webseite und im sich öffnenden Kontextmenü

```
HTTP/1.1 200 OK\r\n
Content-Type: text/html\r\n
Connection: close\r\n
\r\n
<!DOCTYPE html>
<html>
<head>
<title>Arduino WebServer</title>
</head>
<body>
<h3>Arduino Webserver</h3>
<p>Dies ist eine feste Webseite des Webservers.</p>
<p>In naechsten Schritt wird diese Seite flexibel.</p>
</body>
</html>
```

Wie funktioniert der Webserver

Lesen Sie die Kommentare in dem Sketch `WebserverL1.ino`, um zu sehen, was bestimmte Codezeilen tun. Das Listing ist gut dokumentiert. Leider würde dieser Beitrag zu lang, würde man jede Zeile hier ausgiebig erklären. Wenn Sie zur



Bild 3: Im Browser kann man sich auch den HTML-Code der Webseite anzeigen lassen.

auf Seitenquelltext anzeigen. Bild 3 zeigt das Resultat.

Falls Sie daran interessiert sind, den Transfer der Datenpakete im LAN aufzuzeichnen, kann man das Open-Source-Programm Wireshark als leistungsstarkes Dienstprogramm empfehlen. Sie finden es auf der virtuellen DVD im Ordner Utilities. Probieren Sie, Anfrage und Antwort des Arduino-Webservers zu finden. Sie werden staunen, was sich „nebenbei“ sonst noch auf dem LAN abspielt, beispielsweise zwischen dem Router und einem PC. Doch zurück zum Webserver.

Webserver mit SD-Karte

Hat der simple Webserver funktioniert, gehen wir jetzt einen Schritt weiter: Die bisher verwendete Hardware werden verwendet, um einen etwas flexibleren Webserver zu erstellen. Die Webseite ist nicht mehr im Sketch fest „verdrahtet“, sondern wird diesen Mal von der SD-Karte eingelesen. Wenn ein Browser eine Webseite vom Arduino-Webserver anfordert, holt der Arduino die Webseite also von der SD-Karte. Die Dateibezeichnung der Webseite ist mit *index.htm* festgelegt. Nach diesem Dateinamen sucht der Browser, sobald der Webserver eine Anfrage bekommt, in der keine bestimmte Webseite angefragt wird. *Index.htm* ist also ein Standard-Dateiname für den Browser.

Da die Webseite auf der SD-Karte gespeichert werden soll, muss man sie zunächst mit einem Texteditor erstellen oder besser von der virtuellen DVD auf die SD-Karte kopieren. Dabei ist zu beachten, dass die SD-Karte nicht schreibgeschützt ist.

Sie können den Text unten auch abtippen. Es ist möglich, jeden beliebigen Texteditor zu verwenden, beispielsweise den Windows Editor, sogar Windows Notepad. Wenn Sie die Textdatei speichern, geben Sie ihr den Namen *index.htm*. Der Text der Webseite sieht so aus:

```
<!DOCTYPE html>
<html>
<head>
<title>Arduino Webserver mit SD Karte</title>
</head>
<body>
<h3>Arduino Webserver mit SD Karte</h3>
<p>Diese Webseite stammt von der SD-Karte <br>
Man kann den Text leichter auf der SD Karte ändern<br>
und muss den Arduino Sketch nicht ändern.</p>
</body>
</html>
```

Hier gibt es nichts Neues, außer, dass einige Zeilen mit *
* enden. In dem Textabsatz, der mit *<p>* und *</p>* geklammert ist, bewirken die *
* ein Zeilenumbruch. Testen Sie diese Webseite, indem Sie sie auf Ihrem PC in einem Webbrowser öffnen, also im Windows-Explorer anklicken und schauen, ob sie im Webbrowser korrekt angezeigt wird.

Kopieren der Webseite: Sie benötigen einen SD-Kartensteckplatz an Ihrem Computer oder ein Kartenlesegerät, um die SD-Karte zu lesen und zu beschreiben. Stecken Sie die SD-Karte in den Steckplatz am Computer (oder in das Kartenlesegerät) und kopieren Sie die Datei *index.htm* von der Festplatte auf die Micro-SD-Karte. Stecken Sie nun die SD-Karte – ohne Schreibschutz – in den SD-Kartensteckplatz des Ethernet-Shields. (Tipp: Verwenden Sie statt der Dateiendung *.htm* nicht

.html, denn diese Endung mit vier Buchstaben wird von der Arduino-Software nicht unterstützt. Lautet der Dateiname also *index.html*, wird sie auf der SD-Karte nicht gefunden werden.)

Der Arduino-Sketch mit der Bezeichnung *WebserverL2.ino*, der die Webseite von der SD-Karte abruft und an den Browser sendet, finden Sie wie zuvor auf der virtuellen DVD unter *Artikel\ Webserver\ArduinoSketches\ WebserverL2*. Kopieren Sie den Ordner *WebserverL2* nach Laufwerksbezeichnung:*\Users\Benutzername\Documents\Arduino*, also ihr Projektverzeichnis für die Arduino IDE. Aus Platzgründen betrachten Sie *WebserverL2.ino* bitte in der Arduino IDE.

kalen Werte an. Kompilieren Sie den Sketch und laden ihn auf den Arduino UNO. Im seriellen Monitor können Sie verfolgen, ob die Initialisierung von SD-Karte erfolgreich verläuft und die Datei *index.htm* gefunden wird (Bild 4). Surfen Sie mit Ihrem Webbrowser zu der im Sketch eingestellten IP-Adresse. Die von Ihnen erstellte Webseite sollte im Browser angezeigt werden (Bild 5).

Falls etwas nicht funktioniert: Wenn der vorangegangene Sketch läuft, ist das Einzige, was schief gehen könnte, die Initialisierung der SD-Karte und das Auffinden der Datei *index.htm*. Wenn sich die Datei nicht auf der Karte befindet oder nicht den exakten Namen *index.htm* hat, kann der Server die Webseite nicht anzeigen. Öffnen Sie das Fenster des Seriellen Monitors in der Arduino IDE, um die Diagnoseinformationen der SD-Karte anzuzeigen (Bild 4).

Dieser zweite Sketch ist eine etwas veränderte Version des zuvor verwendeten Sketches. Der Sketch initialisiert nun die SD-Karte in der Funktion *Setup()* und sendet Diagnoseinformationen über die serielle Schnittstelle. Anstatt die Webseite zeilenweise aus dem Code heraus zu senden, öffnet dieser neue Sketch nun die Datei *index.htm* von der

SD-Karte und sendet den Inhalt an den Web-Client, also den Webbrowser. Funktioniert das bei Ihnen, können wir nun in einem nächsten Schritt über den Browser eine LED ein- und ausschalten.

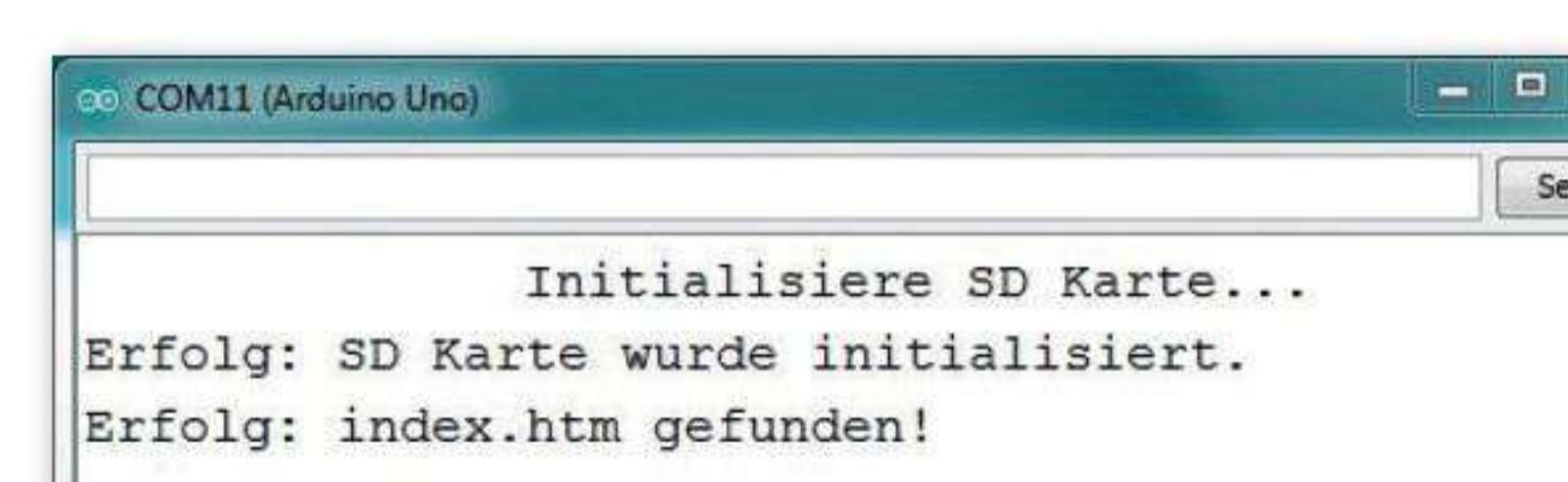


Bild 4: Wurde der Sketch WebserverL2.ino erfolgreich auf den UNO geladen, er erkennt die SD-Karte und die Datei index.htm, sieht die Erfolgsmeldung im Seriellen Monitor der Arduino IDE so aus.

Passen Sie auch in *WebserverL2.ino* die MAC-Adresse, IP-Adresse etc. wie schon zuvor auf Ihre lo-



Bild 5: Der Webserver, der seine Inhalte von der SD-Karte liest.

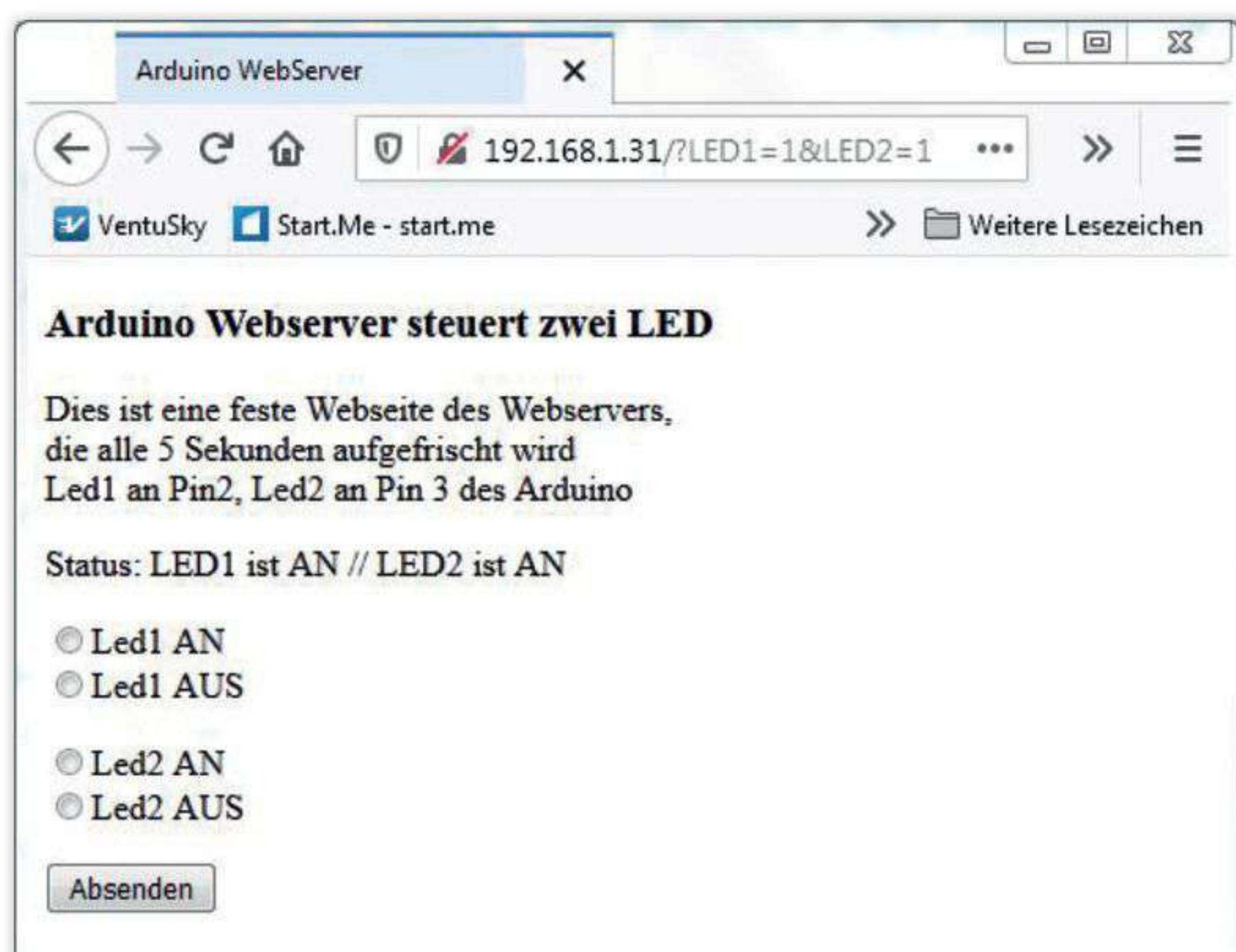


Bild 7: Blick auf den seriellen Monitor der Arduino IDE.

Rot markiert sind die übertragenen Daten des Browsers – dem HTTP-Request – zu sehen: LED1 und LED2 sind eingeschaltet.

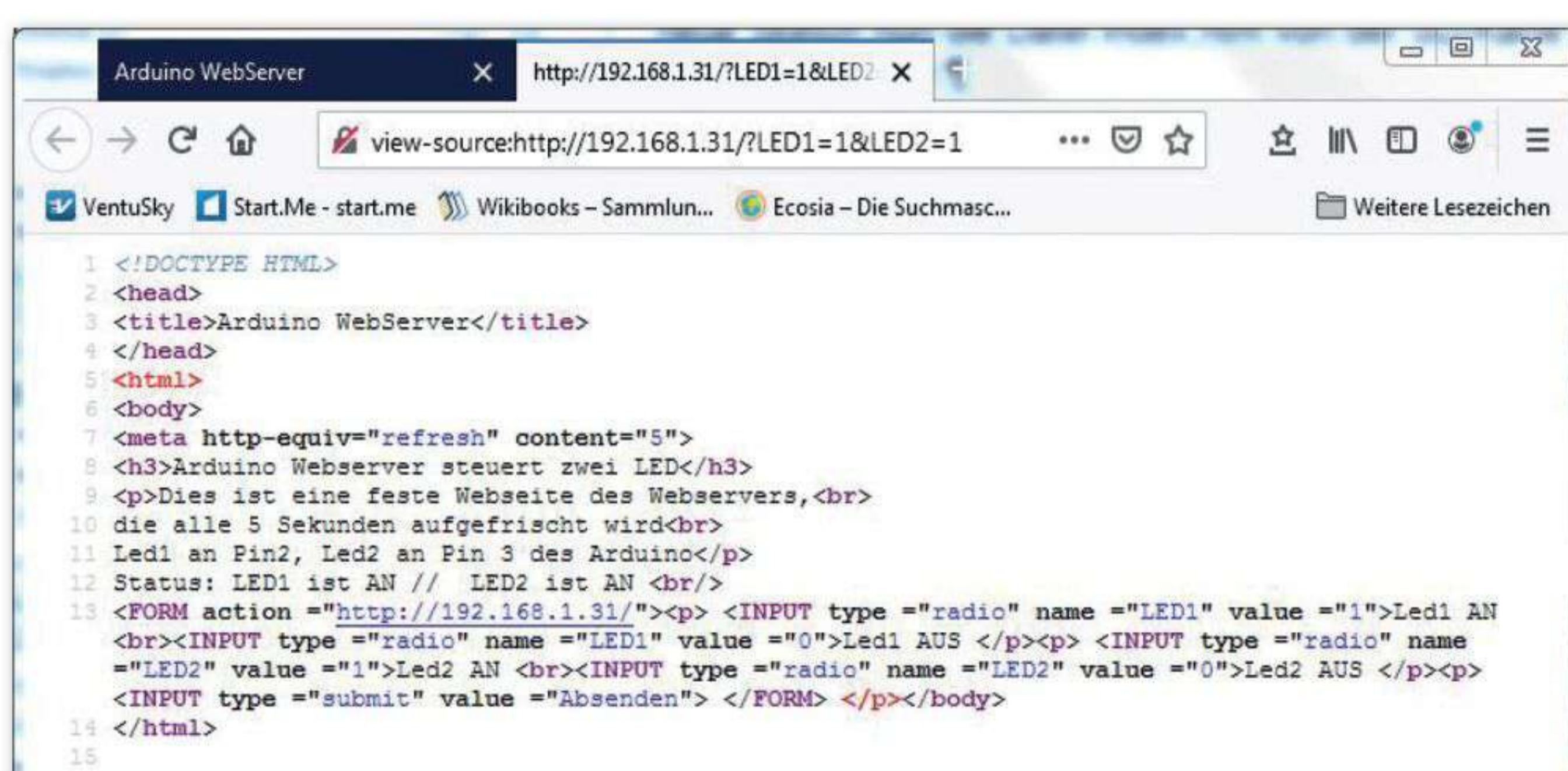


Bild 8: HTML-Code, den der Arduino Webserver an den Browser sendet. Die vier Zeilen unten zwischen <FORM> und </FORM> definieren das Formular mit den beiden Radio-Buttons und der Absenden-Schaltfläche.

weil wir die SD-Karte außen vor lassen. Wir schließen zwei Leuchtdioden über je einen Vorwiderstand an die Arduino Pins 2 und 3 an. Die LEDs sollen leuchten, wenn der Pin2 bzw. 3 High sind. Auf dem Bildschirm wird es später so aussehen wie in Bild 6.

Die Webseite wird hier fest im Sketch erzeugt und ist dennoch in gewisser Hinsicht dynamisch. Es erfolgt alle 5 Sekunden eine Aktualisierung der Webseite, um den aktuellen Status der LED anzuzeigen. Dafür sorgt diese Zeile:

```
client.println("<meta http-equiv=\"refresh\" content=\"5\"");
```

Im Browser kommt schließlich dieses Kommando an:

```
<meta http-equiv="refresh" content="5">
```

Bild 6: Webserver nach WebserverL3.ino, der zwei Ausgänge des Arduino steuert, hier mit Leuchtdioden bestückt.

Bild 7 zeigt den seriellen Monitor mit den Ausgaben, rot markiert die vom Client übermittelte GET-Anweisung mit dem Soll-Status der LEDs. Hier ist interessant, dass der Browser den Status der LEDs in der Adresszeile anzeigt. Durch ein in der Webseite eingebettete Formular, das mit `<form>` eingeleitet und mit `</form>` beendet wird (Bild 8 zeigt die HTML-Tags) und Festlegung auf die GET-Methode, „weiß“ der Browser, dass er – sofern der Benutzer das Formular bedient hat und auf den *Absenden*-Button klickt – diese Informationen zu übermitteln hat. Der Webserver muss „nur“ noch die Zeile mit dem „GET...“ auswerten und die Ausgänge der LEDs schalten. Genauer gesagt, er schaut nach den Zeichenketten „LED1=0“, um die Led1 auszuschalten, „LED1=1“ zum Einschalten dieser sowie „LED2=0“ und „LED2=1“.

Ein wichtiger Hinweis: Wie gehabt sind MAC- und IP-Adresse zu ändern. Die IP-Adresse taucht weiter unten im Sketch noch einmal auf und zwar in dieser Zeile:

```
client.print("<FORM action =\"http://192.168.1.31/\">");
```

Auch hier ändern Sie bitte die IP-Adresse auf die Ihres Arduino-Webservers, sonst funktioniert die Aktualisierung der LED-Status nicht.

Dieser Sketch eignet sich – um ihn besser zu verstehen – gut dazu, ihn in vielfältiger Weise zu modifizieren: Sie können z. B. weitere LEDs einbauen oder andere Änderungen durchführen, um zu sehen, wie sich diese auswirken. Falls Sie so sonderbare Zeichenkombinationen wie \“ finden, wie in obiger Zeile, hat das einen besonderen Sinn: Das Kommando `client.print()` benötigt die Anführungsstriche, um festzustellen, ob es sich um Text oder eine Variable handelt. Um jetzt im Text dennoch ein Anführungszeichen senden zu können, dient die Hilfskonstruktion \“, sie besagt: Sende ein Anführungszeichen, aber interpretiere es nicht als Endemarkierung des `client.print()`-Befehls.

Schalter abfragen mit AJAX

Es ist nicht besonders elegant, eine Webseite alle 5 Sekunden abzufragen, ob sich Veränderungen ergeben haben. Würde man auf eine promptere Aktualisierung von einer Sekunde Wert legen, zeigt sich die

Aktualisierung der Webseite dem Auge als störendes Flackern.

Mit der Programmiersprache JavaScript und dem AJAX-Konzept



Bild 9: Der vierte Webserver in unserer Reihe fragt per JavaScript und AJAX einen Schalter oder Taster ab.

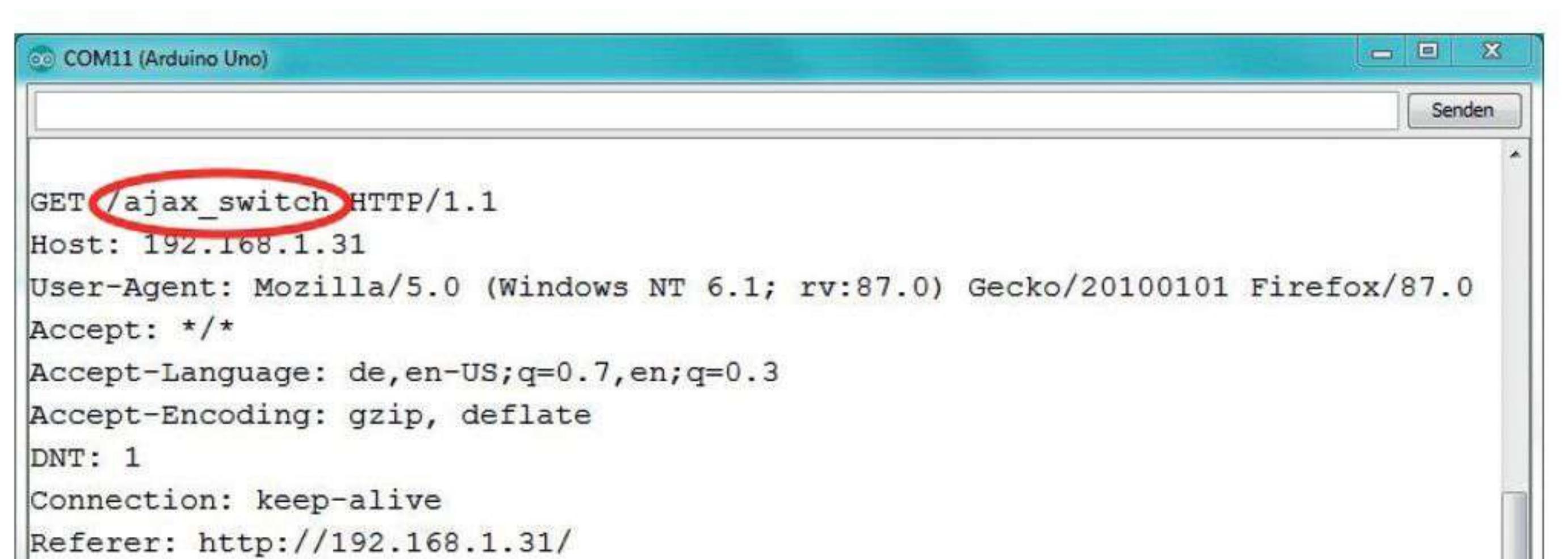


Bild 10: Die rote Markierung beweist: Bei der Übertragung der Daten via AJAX sind die Daten nicht mehr sichtbar.

ist eine elegante Lösung in Reichweite. Bei AJAX handelt es sich nicht um ein Putzmittel, in diesem Fall ist es eine Abkürzung für *Asynchronous JavaScript and XML*. Anders als der Name vermuten lässt, kann man damit nicht nur XML-Daten, sondern allerlei Information übertragen. Im Beispiel ist es ein simpler Text. AJAX ist ein Konzept, das es Webanwendungen ermöglicht, neue Daten vom Server zu erhalten oder dem Webserver für eine Verarbeitung zur Verfügung zu stellen, ohne dass die Seite in Gänze neu geladen werden muss.

Der Sketch *WebserverL4.ino* zeigt ein Beispiel. An Pin A0 des Arduino wird ein Schalter oder ein Taster angeschlossen, dessen Status abgefragt wird (Bild 9). Das AJAX-Schlüsselwort heißt *XMLHttpRequest*. Es handelt sich um ein in modernen Webbrowsern implementiertes Objekt, das wie jedes Objekt im Sinne der Programmierung aus Methoden (Funktionen) sowie Eigenschaften und Ereignissen besteht. Man dem Objekt kann man eine Webseite aktualisieren, ohne dass man sie neu laden muss, Daten vom Server anfordern, nachdem die Seite bereits geladen ist und Daten zu einem Server im Hintergrund senden. Waren bei den bisherigen Beispielen die übertragenen Daten im HTTP-Request sichtbar, bleiben sie hier verborgen, wie man in Bild 10 sieht. Das AJAX-Konzept hier detailliert zu erörtern, würde den Rahmen dieses Beitrages sprengen, daher verweise ich auf eine gute Quelle im Internet [2]. Mag das Ändern eines Textes auf

HTTP: Methoden GET und POST

In den Beispielen haben wir uns bisher auf die GET-Methode fokussiert. Es gibt jedoch weitere Methoden, von denen wir hier eine weitere, die POST-Methode erwähnen wollen. Das Hypertext Transfer Protocol (HTTP) wird hauptsächlich eingesetzt, um Webseiten aus dem World Wide Web in einen Webbrower zu laden. HTTP wurde von der Internet Engineering Task Force (IETF) und dem World Wide Web Consortium (W3C) standardisiert. Die Kommunikation in HTTP zwischen Client (z. B. ein Browser auf PC, Smartphone etc.) und Server werden als Nachrichten bezeichnet, von denen es zwei unterschiedliche Arten gibt: die Anfrage (request) vom Client an den Server und die Antwort (response) als Reaktion darauf vom Server zum Client.

Jede Nachricht besteht dabei aus zwei Teilen, dem Nachrichtenkopf (HTTP-Header) und dem Nachrichtenrumpf (Body). Der Nachrichtenkopf enthält Informationen über den Nachrichtenrumpf wie etwa verwendete Kodierungen, damit dieser vom Empfänger korrekt interpretiert werden kann. Der Nachrichtenrumpf enthält schließlich die Nutzdaten.

Zur Übertragung wird über TCP auf dem Standard-Port 80 des HTTP-Servers eine HTTP-GET-Anfrage gesendet.

```
GET /HTTP/1.1\r\n
\r\n
\r\n
```

einer Webseite bisher als nicht spektakulär empfunden werden, derjenige sollte sich das nächste und letzte Beispiel eines Arduino-Webservers anschauen, dessen prima Grafikanzeige eines analogen Wertes ich auf [3] fand.

Webserver mit Grafikanzeige

Was heute technisch möglich ist, verdeutlicht das letzte Beispiel eines Webservers (Bild 11). Ein an Arduino Pin A0 angeschlossenes Poti, dessen Wert ausgelesen wird, zeigt dieses Beispiel in bester Grafikqualität auf dem Browser an.

Das Beispiel verwendet die SD-Karte, somit kopieren Sie von der virtuellen DVD die Datei *index.htm* aus dem Ordner *\Artikel\Webserver\WebserverL5\Webseite* auf die SD-Karte. Den Ordner *\Artikel\Webserver\WebserverL5* haben Sie evtl. schon beim WebserverL1 in das Arduino-Projektverzeichnis kopiert. Vergessen Sie bitte nicht, MAC und IP-Adresse anzupassen!

Bild 11: JavaScript und AJAX ermöglichen ansprechende Grafik und Anzeige der Werte fast in Echtzeit.



Das ist die Anfrage (request) eines Webrowsers an den Webserver, den Inhalt einer Webseite zu senden. Der Webserver liefert eine Antwort (response) wie diese hier von einem Apache-Server unter Unix:

```
HTTP/1.1 200 OK
Server: Apache/1.3.29 (Unix) PHP/4.3.4
Content-Length:
Content-Language: de
Connection: close
Content-Type: text/html
```

(... es folgt der Inhalt von *index.htm*)

Da der Dateiname in der Anfrage nicht angegeben wurde, liefert der Server als Standard die Datei *index.htm* oder *index.html* zurück. Der Server sendet eine Fehlermeldung sowie einen Fehlercode zurück, wenn die Information aus verschiedenen Gründen, etwa, weil die Datei nicht existiert, nicht gesendet werden kann. Es wird bei Erfolg der Statuscodes 200 OK gesendet.

HTTP-GET ist die gebräuchlichste Methode. Mit ihr wird eine Ressource (zum Beispiel eine HTML-Datei) vom Server angefordert. Als Argumente können auch Inhalte zum Server übertragen werden. Das sind im Beispiel des WebserverL3 die Informationen über den Zustand der Leuchtdioden *\?LED1=0&LED2=1*. Dabei gilt: Die Argumente werden mit der Zeichenfolge *\?* eingeleitet, es folgt Argument1 (LED1=1), das zweite Argument wird durch das *&*-Zeichen vom ersten getrennt, darauf folgt Argument

zwei (LED2=1) und so fort. Auf diese Weise überträgt man mehrere Argumente in einer Zeile.

HTTP-POST schickt (fast) unbegrenzte Mengen an Daten zur weiteren Verarbeitung zum Webserver, sie werden als Inhalt der Nachricht übertragen (*body*) und bestehen beispielsweise aus Name-Wert-Paaren (z. B. *LED1=1*). So entstehen neue Ressourcen wie Dateien oder Datenbanken auf dem Server oder es werden bestehende Informationen verändert. Quelle: Diverse und Wikipedia.org. Ein Beispiel sehen Sie unten, hier wurde für die Übertragung der beiden LEDs die Methode POST gewählt:

```
POST /HTTP/1.1
Host: 192.168.1.31
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: text/html,application/xhtml+xml...
Accept-Language: de,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 13
Origin: http://192.168.1.31
DNT: 1
Connection: keep-alive
Referer: http://192.168.1.31/
Upgrade-Insecure-Requests: 1
```

LED1=1&LED2=1
// hier die Info bei POST

```
HTTP/1.1 200 OK
Content-Type: text/html
Connection: close
```

bewegt sich der Zeiger brav analog zur Stellung des Potentiometers an A0.

Zum Abschluss

Mit AJAX kann man eine Menge anstellen: Ein SWR-Meter entwickeln, um die SWR-Anzeige des Funkgerätes über das LAN verfügbar werden zu lassen. Die Stellung des Rotors wäre eine andere Idee. Was fällt Ihnen ein, das Sie gern programmieren würden?

Wer nicht in die hohe Programmierung mit JavaScript und AJAX einsteigen will, hat zumindest viel weniger Arbeit, wenn er sich mit der Darstellung von Ziffern auf der Webseite begnügt. Das sollten diese Beispiele demonstrieren. Viel Spaß bei der Durcharbeit der fünf Arduino-Webserver.

Literatur/Verweise:

- [1] Arduino-Homepage: <https://www.arduino.cc/>
- [2] Infos zu AJAX: <https://wiki.selfhtml.org/wiki/JavaScript/Ajax>
- [3] Analoge Anzeige: <https://github.com/Mikhus/canvas-gauges>
- [4] JavaScript und AJAX-Kurs: <https://www.javascript-kurs.de/ajax-einfuehrung.htm>

Michael Wöste, DL1DMW

Schalten mit PC, aber ohne Löten und Programmieren: USB-RELAIS

Nicht jeder möchte das Programmieren lernen, vielleicht aber auf die Schnelle vom PC gesteuert ein Relais schalten oder den analogen Wert eines Sensors einlesen. Ein kleiner Arduino Sketch macht aus einem Arduino UNO eine Reihe von „Schalter“, die man über USB und mit einfachen Kommandos anspricht.

Besteht die Aufgabe darin, mittels PC auf die Schnelle ein Relais, eine LED-Beleuchtung, die PTT des Funkgerätes oder irgendetwas anderes digital zu steuern – also etwas ein- bzw. auszuschalten, steht die altbewährte parallele oder serielle Schnittstelle (LPT, RS232) am PC heute nicht mehr zur Verfügung. Mit beiden konnte man das leicht realisieren. Als Alternative bietet sich heute die flexible, schnelle und allgegenwärtige, aber komplexer anzusprechende USB-Schnittstelle an. Auch für USB gibt es fertige Lö-

Arduino UNO flashen

Damit der Arduino UNO die Kommandos erkennt und ausführt, muss er einmal mit dem Programm bestückt werden. Man kann auch sagen: das Programm wird auf den Arduino aufgespielt. Dazu dient die Arduino-Entwicklungsumgebung (Arduino IDE), die Sie auf dem PC installieren und starten. Schließen Sie nun den Arduino UNO an den PC an und warten einige Sekunden, bis Windows den Treiber automatisch installiert hat. Im Menü *Werkzeuge* wählen Sie unter *Board* nun *Arduino UNO* aus und unter *Port* den virtuellen Com-Port, den Windows dem UNO zugeordnet hat. Nun laden Sie den Sketch *USB-Relais.ino*, den Sie zuvor in das Projektverzeichnis der Arduino IDE kopiert haben. Den Sketch finden Sie im Ordner *\Artikel\USB-Relais* der virtuellen DVD zum Download. Als letzten Schritt übertragen Sie den Sketch auf den UNO (Menü *Sketch/Hochladen*). Rufen Sie anschließend unter *Werkzeuge/Serieller Monitor* diesen auf und geben den ersten Befehl ein.

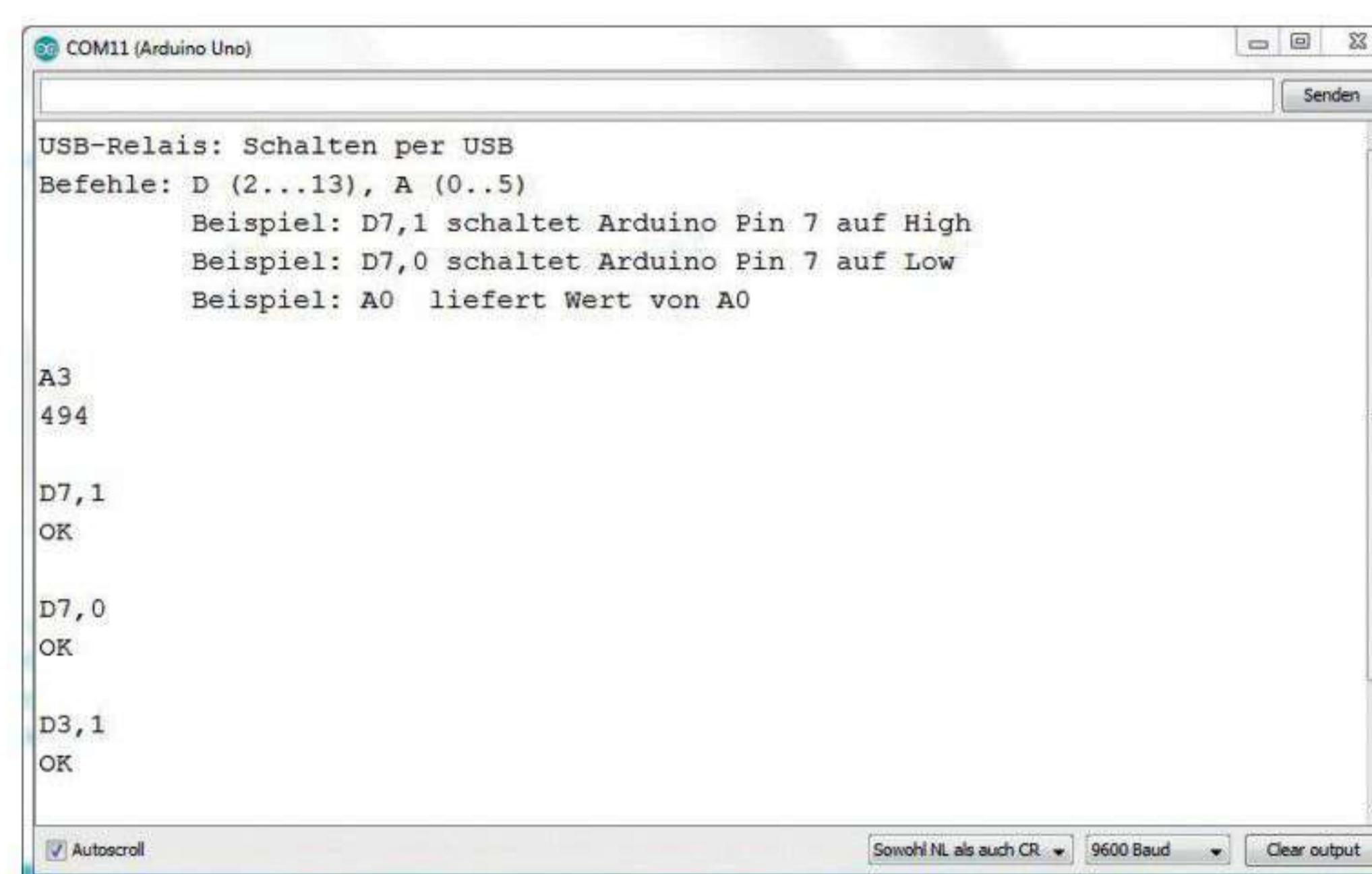


Bild 1: Serieller Monitor der Arduino IDE: Eingabe von Kommandos zur Steuerung.

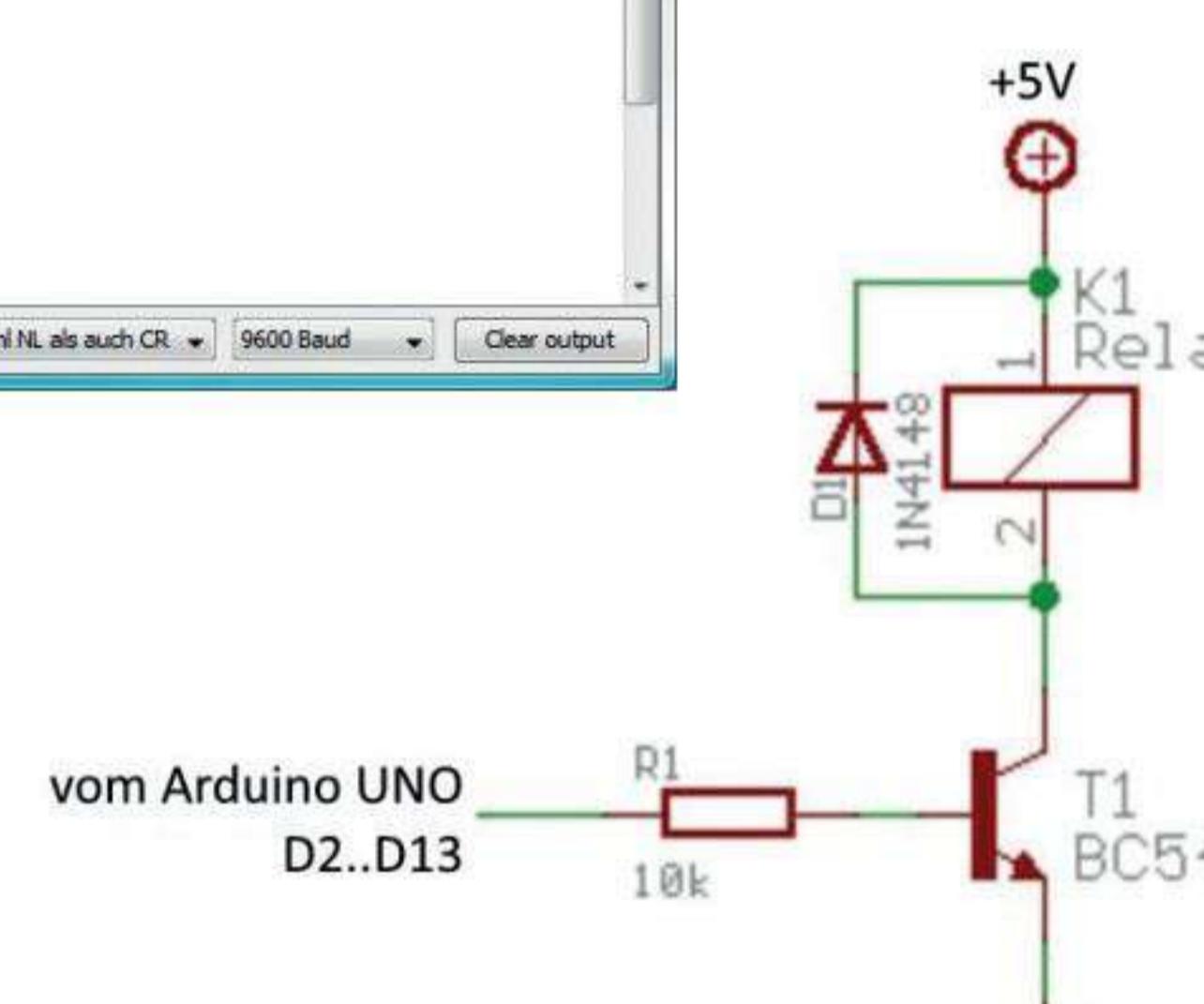


Bild 2: So steuert man beispielsweise ein 5-Volt-Relais mit dem Arduino UNO an.

sungen, doch die sind oft, hat man den Wunsch, eine Lösung zeitnah bereit zu stellen, nicht parat. Als Lösung bietet sich – wieder einmal – ein Arduino an, der „von Haus aus“ bereits über eine USB-Schnittstelle programmiert und bedient wird. Sie mögen nun einwenden, dass man den Arduino programmieren muss. Das ist zwar richtig, aber in diesem Fall bereits erledigt. Um das USB-Relais in Betrieb zu setzen, muss der vorhandene Sketch *USB-Relais.ino* einmal auf den Arduino UNO kopiert werden. Von nun an stehen zwei Befehle zur Verfügung, die einfach zu merken sind und für die man nicht programmieren lernen muss. Aus einem Arduino UNO wird also eine Steuerung für eines oder mehrere Relais, oder, wenn Sie so wollen, die Schnittstelle zwischen einem zu steuernden Gerät und Ihrem Personal Computer.

Der Arduino UNO weist einige Prozessorspins auf, die auf der Platine

mit 2...13 bezeichnet sind und fünf weitere, die mit A0 bis A5 benannt wurden. An die digitalen Ausgänge D2 bis D13 schließen Sie beispielsweise bis zu elf Leuchtdioden, Relais oder anderes an. Mit einem kurzen Kommando schalten Sie diese einzeln ein- bzw. aus. Bei den mit A0 bis A5 bezeichneten Anschlüssen auf der Platine handelt es sich um analoge Eingänge, über die Sie beispielsweise den Spannungswert eines Potentiometers oder eines analogen Sensors im Bereich von 0...5V einlesen.

Und wie übermittelt man der Platine ein Kommando? Das ist einfach: Da der Arduino UNO, ist er mit dem PC verbunden – im PC-Betriebssystem Windows einen

virtuellen COM-Port belegt, bietet es sich an, diesen Weg für das Absetzen der Kommandos zu nutzen. Dazu benötigen Sie entweder den in der Arduino IDE vorhandenen seriellen Monitor oder ein beliebiges Terminalprogramm (HTerm, HyperTerm, YetAnotherTerminal, PUTTY...).

Nehmen wir einmal an, Sie haben ihren Arduino UNO bereits mit dem Programm bestückt und möchten das erste Relais schalten. Nehmen wir weiter an, dass es an Pin 2 des Arduino UNO angeschlossen ist.

Mit dem Kommando D2,1 schalten Sie das Relais ein und mit D2,0 wieder aus. Im seriellen Monitor tippen Sie die Kommandos in die Eingabezeile ein und betätigen dann die Return-Taste. Der Buchstabe D bedeutet: Schalte digital, die 2 entspricht dem Aufdruck auf der Arduino UNO-Platine und die 1 steht für den Pegel HIGH. Dem

entsprechend schaltet D2,0 den Anschluss zwei auf LOW, das Relais also wieder aus. Noch ein Beispiel: D13,1 schaltet Anschluss 13 auf HIGH, D9,0 den Anschluss 9 auf LOW.

Es bleiben noch die Anschlüsse A0 bis A5. Die sind als analoge Eingänge geschaltet und werden mit ihrer Bezeichnung abgefragt. Der UNO liefert prompt dessen analogen Wert. Beispiel: A0 fragt den Anschluss A0 ab und liefert dessen Analogwert, A4 den vom Anschluss A4. Der A-Befehl liefert Werte von 0...1023 zurück. Das entspricht dem Spannungsbereich von 0 bis 5 Volt. Bitte nach einem Kommando nicht vergessen, die Return-Taste zu drücken! Der Arduino UNO wartet darauf und schaltet dann. Probieren Sie es aus! Möchten Sie den erhaltenen Wert in eine Spannung umrechnen, benutzen Sie diese Formel: $\text{Volt} = (5V / 1024) * \text{gemessener Wert}$.

Literatur/Verweise:

- [1] Arduino IDE: <https://www.arduino.cc/en/software>

Michael Wöste, DL1DMW

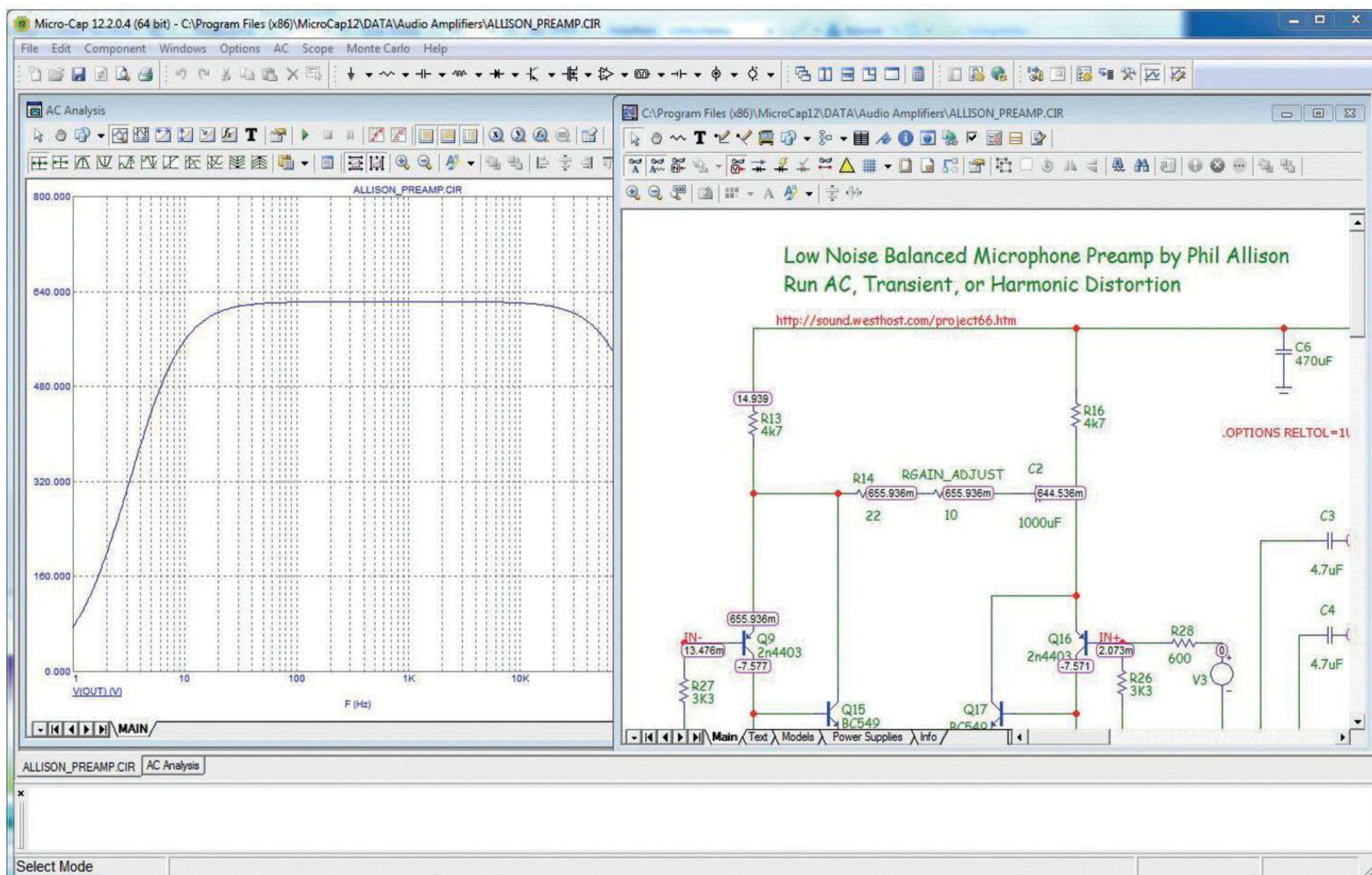


Bild 1: MicroCAP 12 zeigt links eine Filterkurve, rechts den Schaltplan des NF-Verstärkers.

Monte-Carlo-Analyse mit MicroCAP12

MicroCAP-Basics – eine Einführung

Neueinsteiger sollten etwas Grundwissen mitbringen, damit es mit der Simulation – das Benutzerhandbuch spricht präziser von Analysen – klappt. Das über 200-seitige, englischsprachige Handbuch (User-Guide) ist als erste Lektüre sehr zu empfehlen. Das detailliertere Reference-Manual zählt fast

1100 Seiten! Dennoch lohnt zum Einstieg eine kurze Einführung auf wenigen Seiten: Alles beginnt mit der Platzierung der Komponenten auf dem Schaltplan-Arbeitsblatt. Für Bauteile wie Widerstand, Kondensator, Spule, Diode etc. stehen in der Funktionsleiste entsprechende Symbole bereit. Ebenso findet man derlei Bauteile unter *Analog Primitives* links im MicroCAP-Fenster wie auch im Menü *Component / Analog*

Primitives / Passiver Components. Legt man ein Bauelement mit der Maus auf der Arbeitsfläche ab, öffnet sich der Eigenschaften-Dialog und ein Cursor blinkt zur Eingabe des Wertes. Hierbei ist die Syntax strikt zu beachten:

Eingabe	Bedeutung
Meg	Mega
k	Kilo
M oder m	Milli
u	Micro (μ)
p	Pico

Ein großes M bedeutet also Milli und nicht, wie man vermuten könnte Mega! Reelle Zahlen werden nicht mit dem Komma, sondern mit dem Punkt geschrieben (statt 4,7k also 4.7k), Sie können aber auch 4k7 sch-

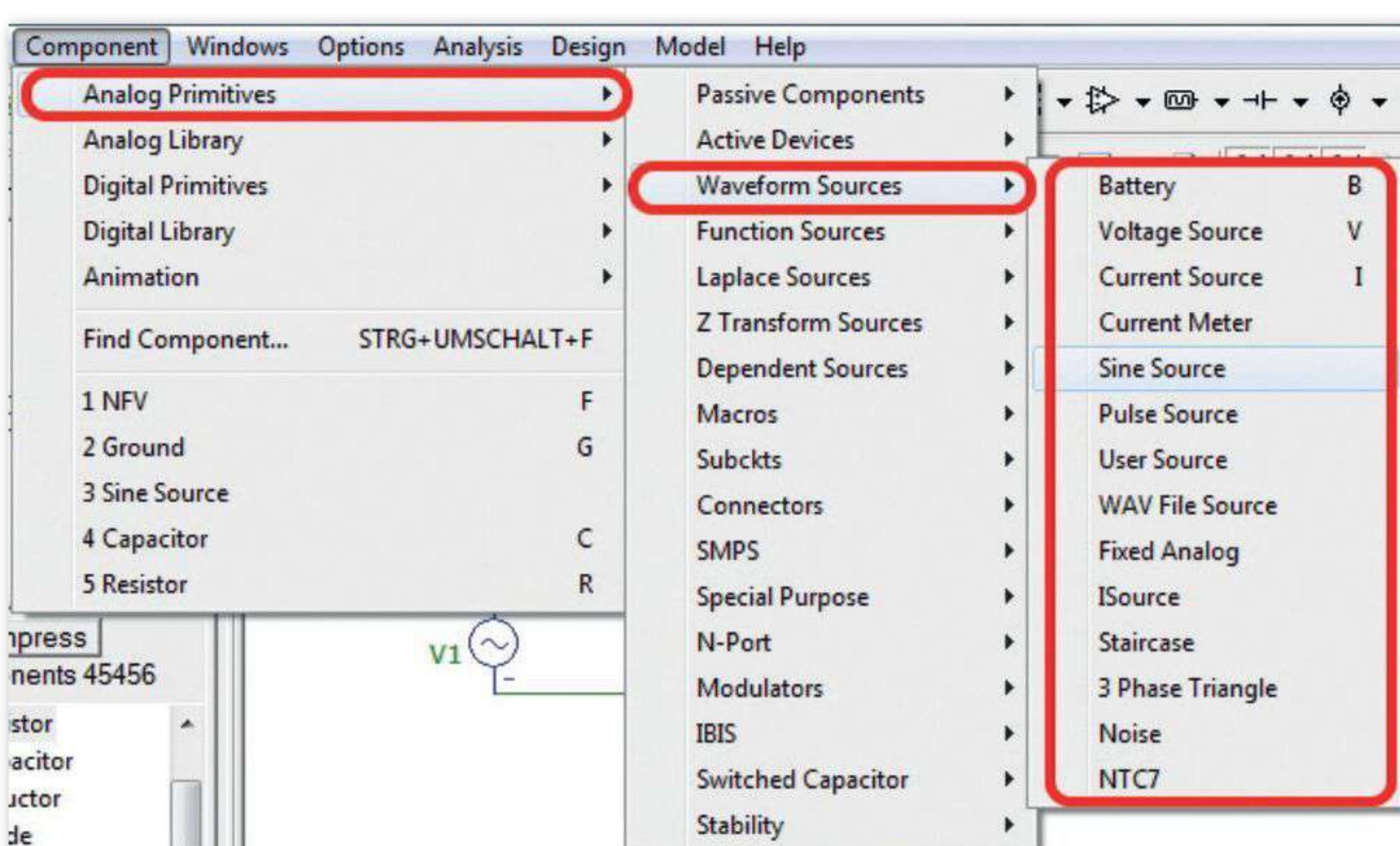


Bild 2: Wahl einer Spannungs- oder Stromquelle über das Menü Components / Analog Primitives.

MicroCAP ist das professionelle – und inzwischen für alle frei gegebene – Simulationsprogramm für analoge und digitale Schaltungen auf der Basis von SPICE. Neben einigen Basics, die bei der Nutzung des Programms zu beachten sind, soll gezeigt werden, wie sich Bauteiltoleranzen auf die Funktion einer Schaltung simulieren lassen.

reiben. Sie verlassen den Dialog mit der Taste <ENTER> oder über den Button OK. Fügen Sie über *Components / Analog Primitives / Waveform Sources* eine Spannungsquelle oder einen Sinusgenerator hinzu.

Wir bauen für eine erste Simulation einen einfachen Tiefpass aus Widerstand (4k7), Kondensator (1u) und einer Sinusquelle in Serie. Geben Sie bei letzterer unter *Value* einen Wert von 10 kHz an.

Jede Schaltung benötigt ein Massesymbol (*Ground*), ohne das eine korrekte Simulation nicht möglich ist. Nun verdrahten Sie die Bauelemente miteinander. Aktivieren Sie zuerst im Menü *Options / View* die Anzeige *Pin Numbers* und *Knotennummern (Node Numbers)*. Damit erkennen Sie, ob Bauelemente korrekt mit dem „Draht“ verbunden sind. Knotennummern werden nach der Simulation bzw. Analyse bedeutsam. Schalten Sie nun mit STRG+W (oder über das Drahtsymbol in der Funktionsleiste) in den Drahtmodus (*wire mode*) und verbinden die Bauelemente untereinander. Das geschieht intuitiv. Unsere simple Schaltung könnte auf Ihrem Monitor etwa so aussehen (Bild 3).

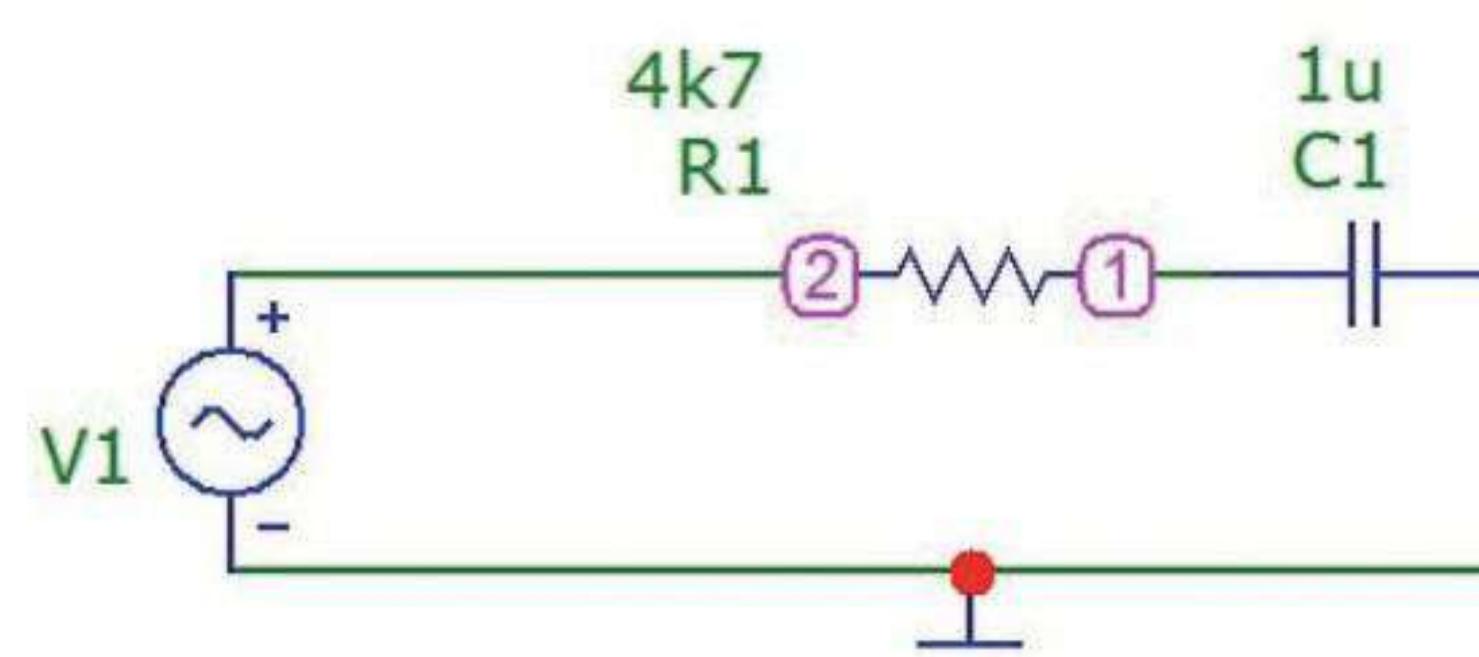


Bild 3: Das Schaltbild des einfachen RC-Gliedes als Tiefpass.

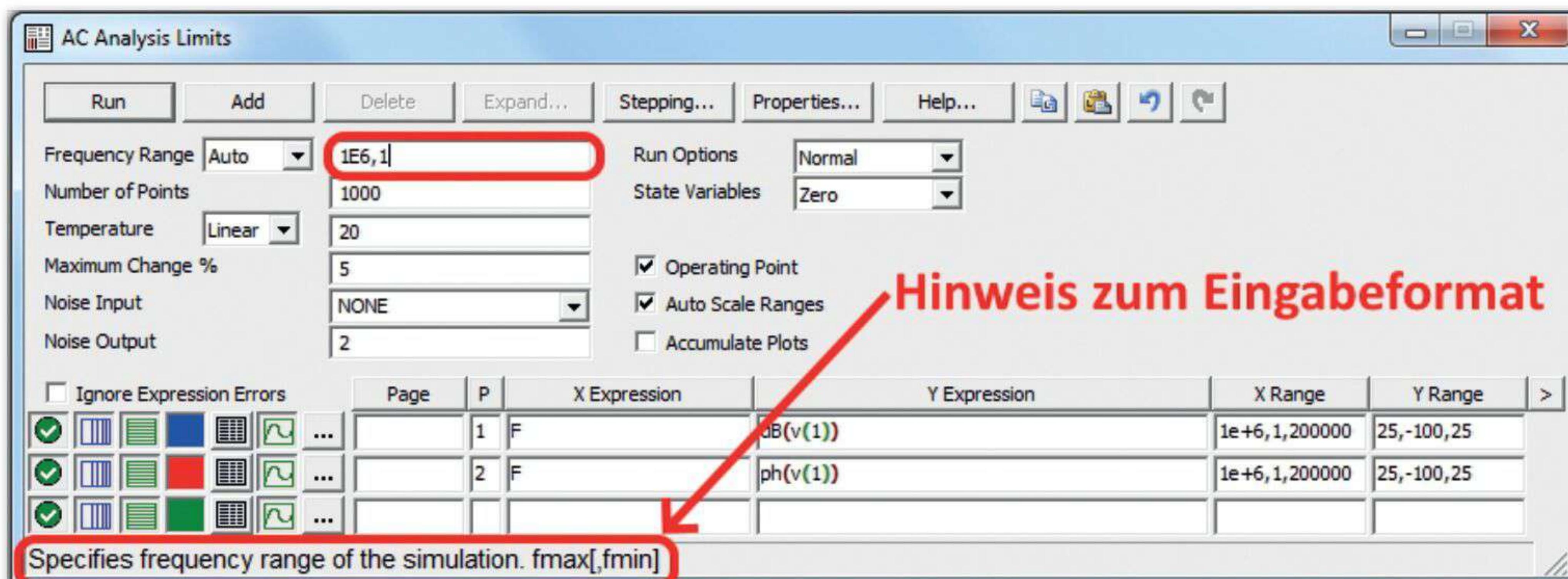


Bild 4: Dialog zur AC-Simulation. Die Zeile unten gibt wichtige Hinweise zur Eingabesyntax.

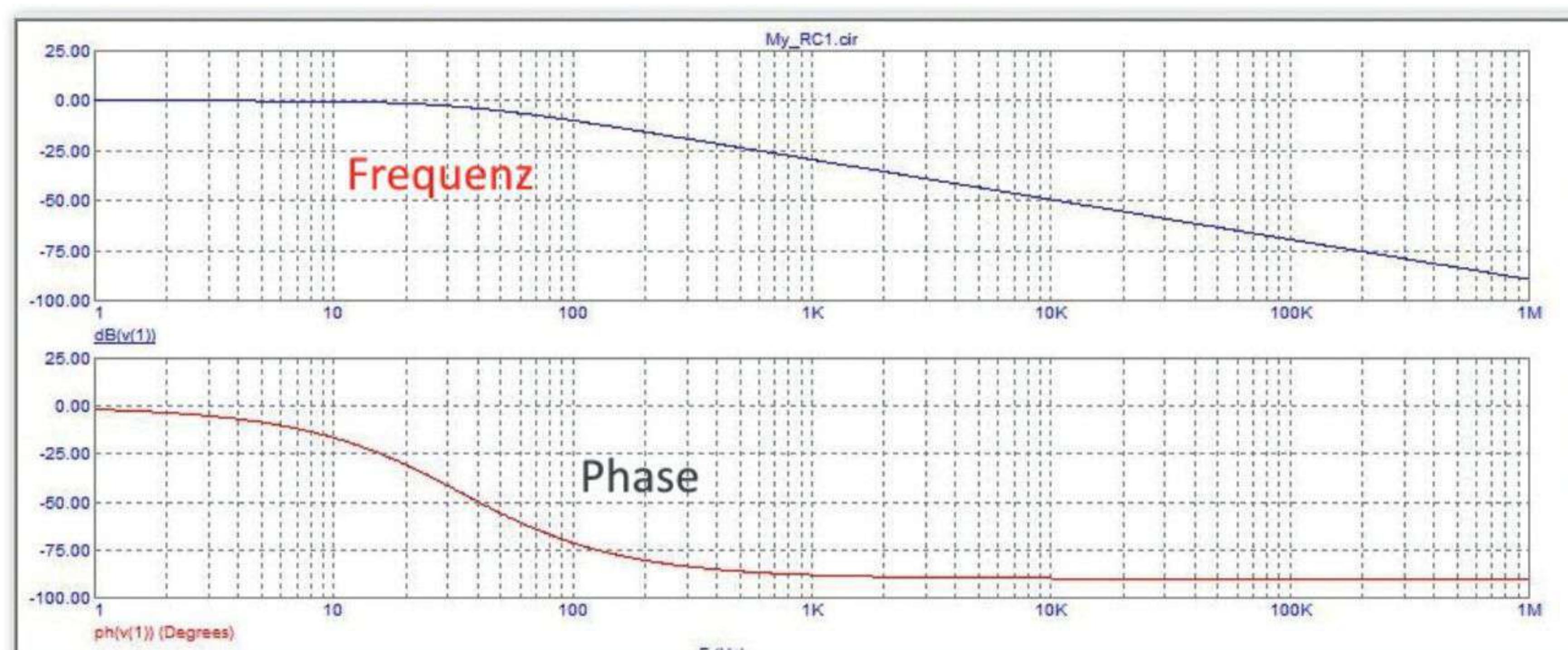


Bild 5: Frequenz und Phase des Tiefpass-Filters in Bode-Darstellung.

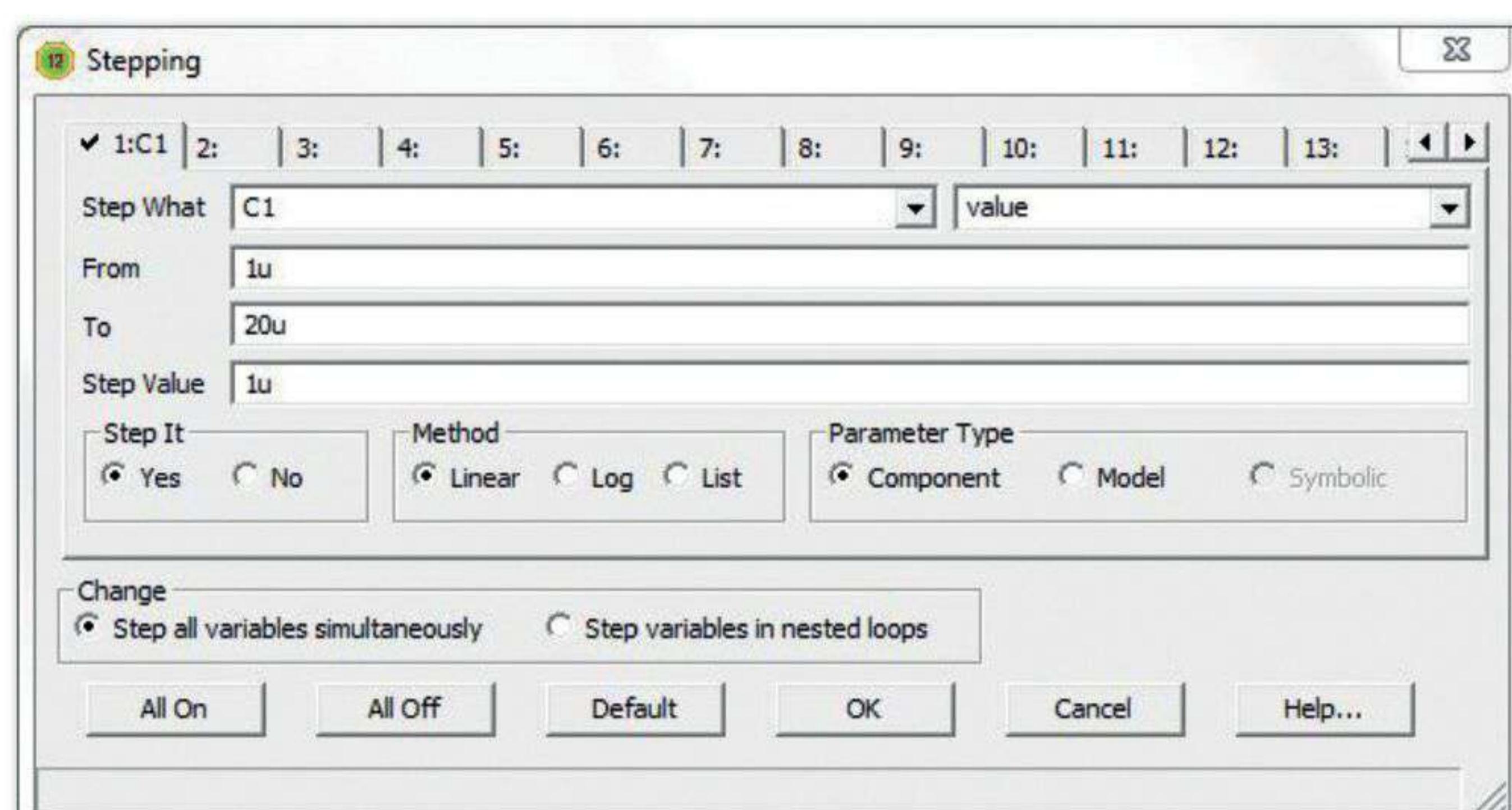


Bild 6: In dem Stepping-Dialog legt man eines oder mehrere Bauelemente fest, deren Wert bei der Analyse verändert werden.

Möchten Sie den Wert bzw. die Eigenschaften eines Bauteils nachträglich ändern, gelangen Sie in das Eigenschaftenfenster mit einem Doppelklick auf das Bauelement.

Starten Sie aus Interesse eine Transienten-Analyse (Menü *Analysis / Transient*). Das entspricht der Darstellung eines Oszilloskops. Belassen Sie alle Parameter, wie sie das Programm eingetragen hat und starten mit *Run*. Erwartungsgemäß zeigt die Analyse

eine Sinusschwingung. Beachten Sie bitte: Die Menüleiste weist neue Menüs wie *Transient*, *Scope* und *Monte Carlo* auf, die weitere Funktionen bieten. Beenden Sie die Analyse durch Schließen des Fensters oder klicken im Menü *Transient* auf *Exit Analysis* oder betätigen die Funktionstaste F3.

Wir können nun, um den Effekt des Tiefpasses zu erkennen, eine erste AC-Analyse starten. Im Menü *Analysis / AC* oder mit ALT+2 öffnet sich der Dialog zur Eingabe der Simulationsparameter (Bild 4). Bei der Wechselstrom-Simulation des Tiefpasses ist die Vorgabe des Frequenzbereiches maßgeblich. Dabei muss im Feld *Frequency Range* die Syntax eingehalten werden: Zuerst wird die höhere Frequenz in Hz, gefolgt von einem Komma die niedrige Frequenz in Hz angegeben, wie Bild 4 verdeutlicht (1E6 Hz = 1000000 Hz = 1 MHz). Ein Klick bei *Auto Scale Ranges* bewirkt eine automatische Skalierung beider Achsen des Bode-Plots.

Gehen Sie mit der Maus auf Entdeckungskurs: Ändern Sie die X-Skalierung von logarithmisch zu linear, verändern Sie die Form der Ausgabe vom Bode-Plot zur Polardarstellung oder einem Smithdiagramm oder verändern die Zeichenfarbe des Graphen. Schauen Sie sich die Parameter der Spalten X- und Y-Expressions an, die erscheinen, sobald Sie in einem der Eingabefelder die rechte Maustaste betätigen. Erkunden Sie die Möglichkeiten des Kontextmenüs.

In den Spalten X- und Y-Expression trägt man die zu berechnenden Formeln ein. Hier hat es das Programm bereits erledigt. Die X-Achse enthält nur ein „F“ und damit wird auf der X-Achse die Frequenz aufgetragen. Die Y-Achse ist mit *dB(v(1))* belegt. Die „1“ bezieht sich dabei auf die Knotennummer (*node number*), die Sie im Schaltplan sehen. Am Ende starten Sie die Analyse mit *Run*.

Das Resultat der Analyse zeigt Bild 5. Oben sehen Sie den Frequenzgang, darunter den Phasengang. Möchten Sie Parameter der Analyse verändern? Mit F9 öffnet sich der Dialog erneut. Weitere Funktionen bietet das Menü *AC*.

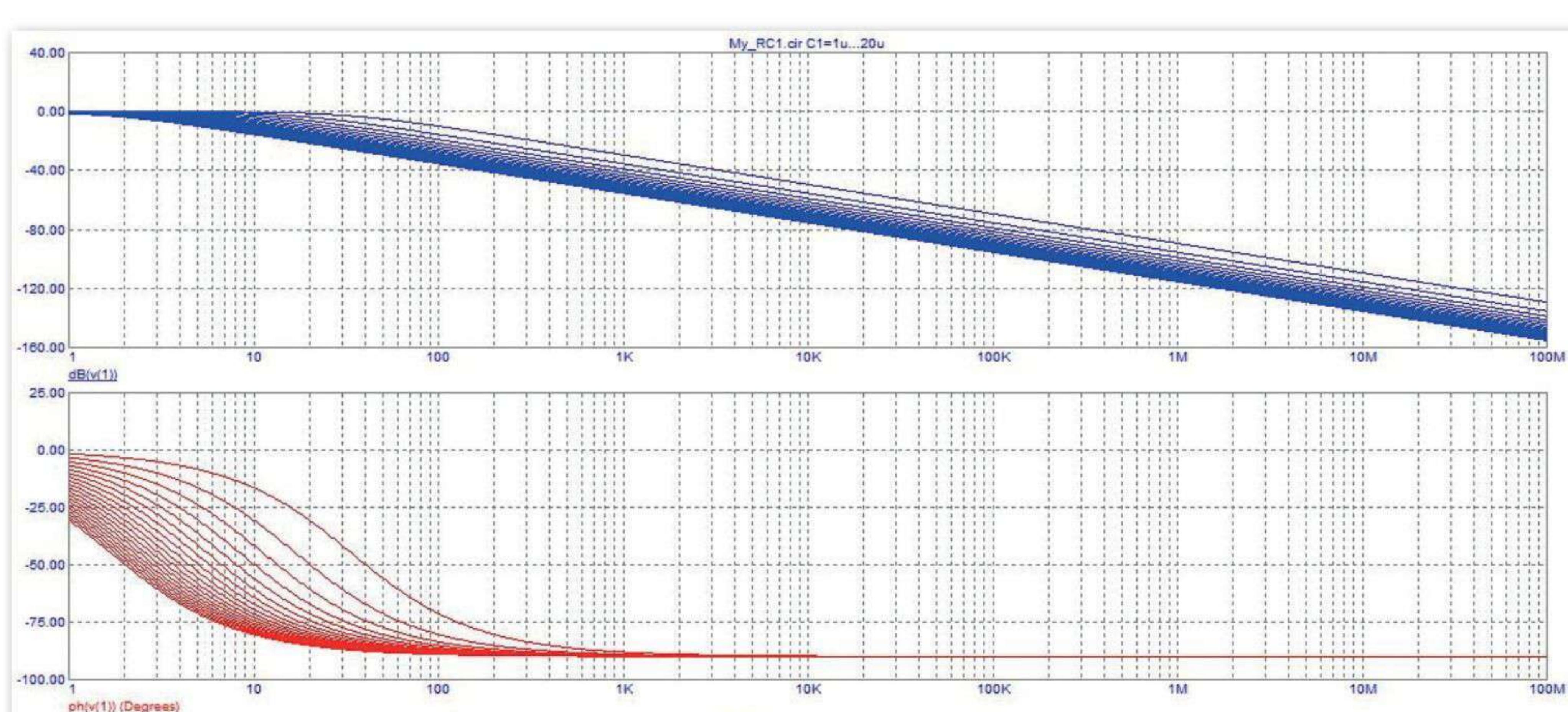


Bild 7: Die Analyse bei Variation von C1 zeigt die sich ergebende Kurvenschar für Werte zwischen 1 μ F und 20 μ F.

Stepping

Im Parameter-Dialog zu den Analysen gibt es oben einen Button *Stepping* (Bild 4). Hier bietet sich die Möglichkeit, Bauteilewerte in der Simulation zu verändern. Wir wollen den Wert des Kondensators von $1\mu\text{F}$ zwischen $1\mu\text{F}$ und $20\mu\text{F}$ variieren und das Ergebnis grafisch ausgeben lassen. In dem Stepping-Dialog (Bild 6) wählen Sie unter *Step What* den Kondensator C1 aus und tragen die Werte *From* und *To* wie in Bild 6 ein. Nach einem Klick auf *OK* erscheint ein weißes Analysefenster. Nichts passiert. Im Menü *AC* starten Sie nun *Run* oder drücken F2. Danach sollte es etwa

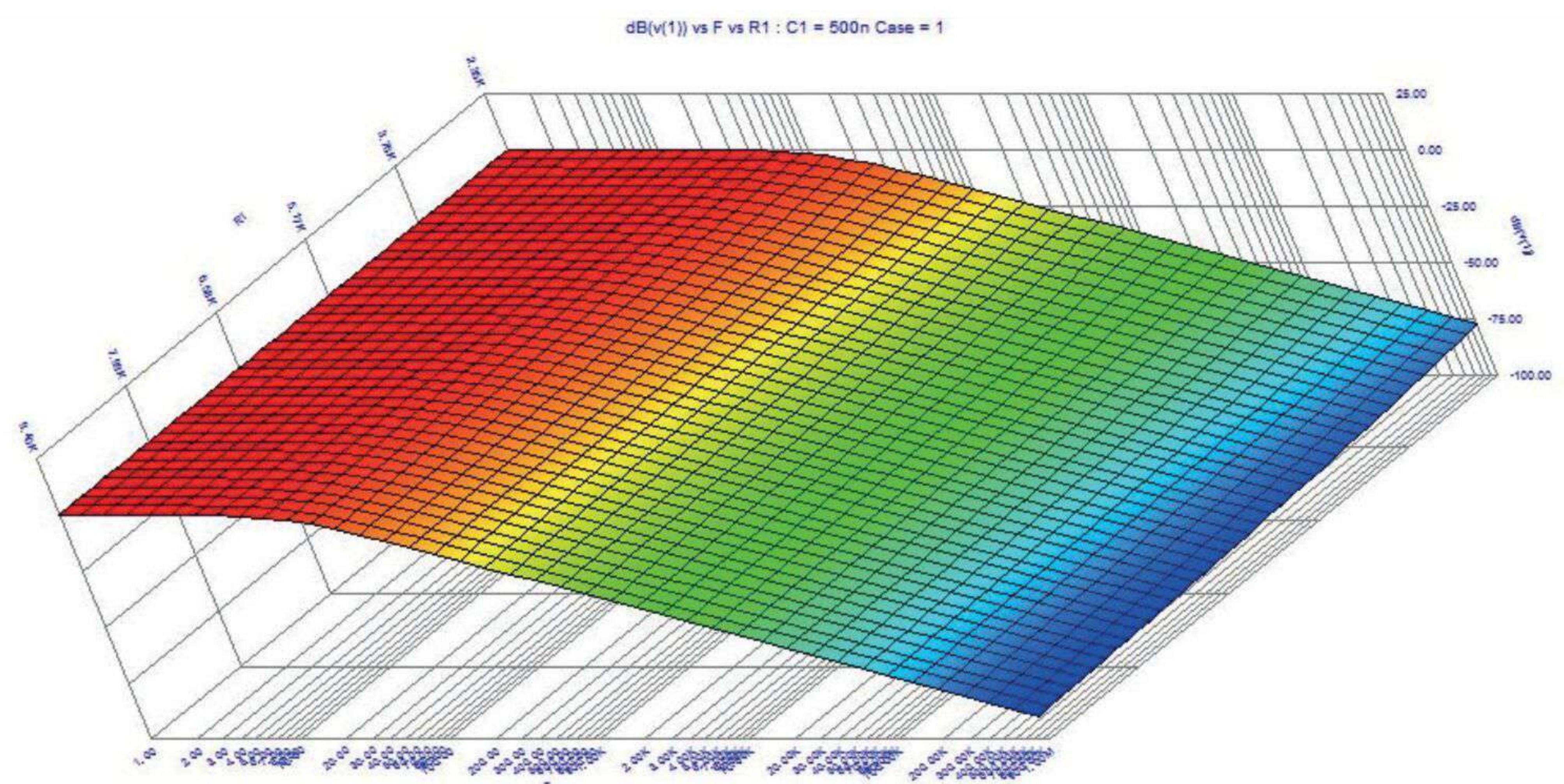


Bild 8: Variiert man die Werte von R und C in einer Analyse, erzeugt MicroCAP auf Wunsch einen dreidimensionalen Plot.

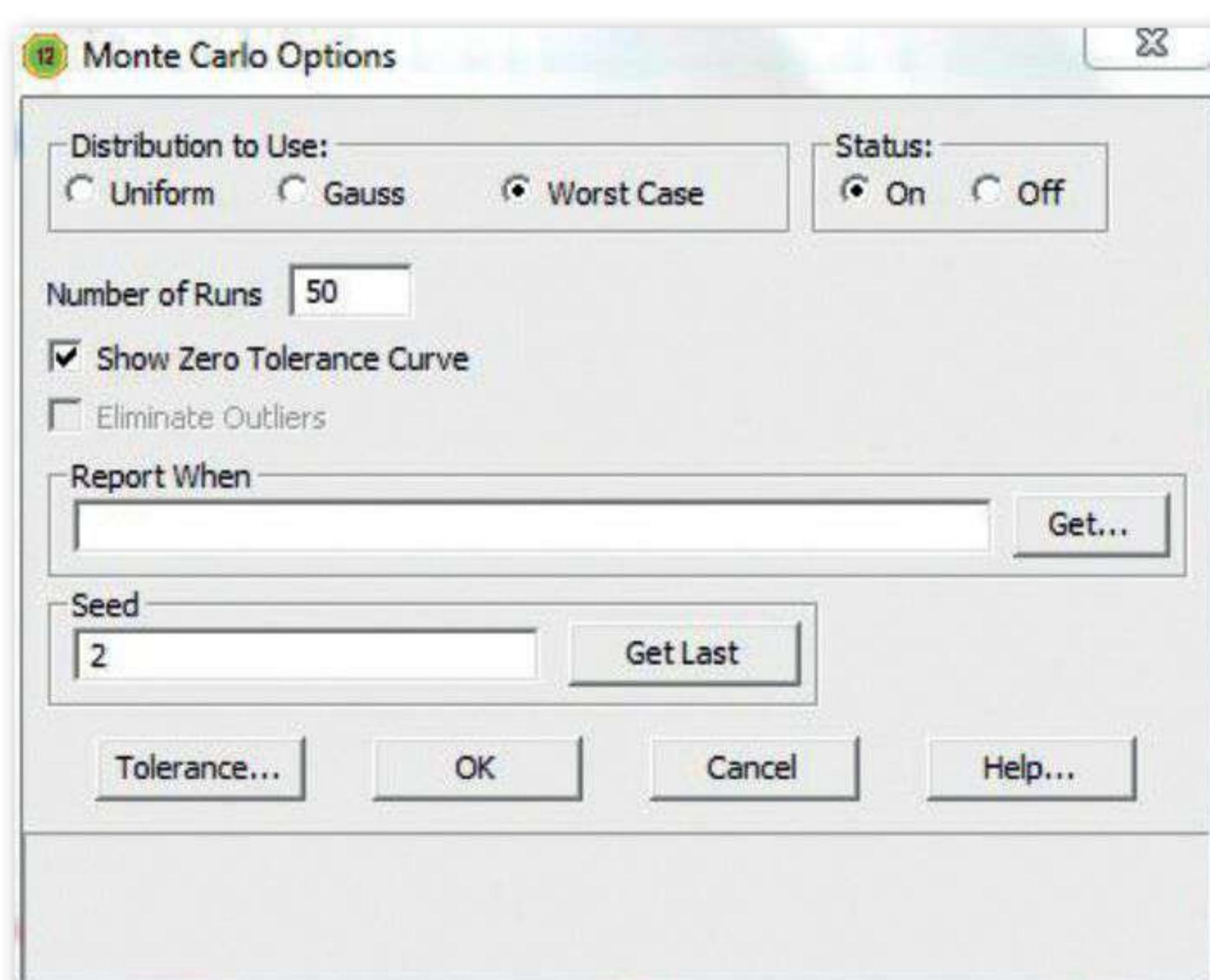


Bild 9: Der Monte-Carlo-Optionen-Dialog. Für unser Beispiel übernehmen Sie bitte die Einstellungen.

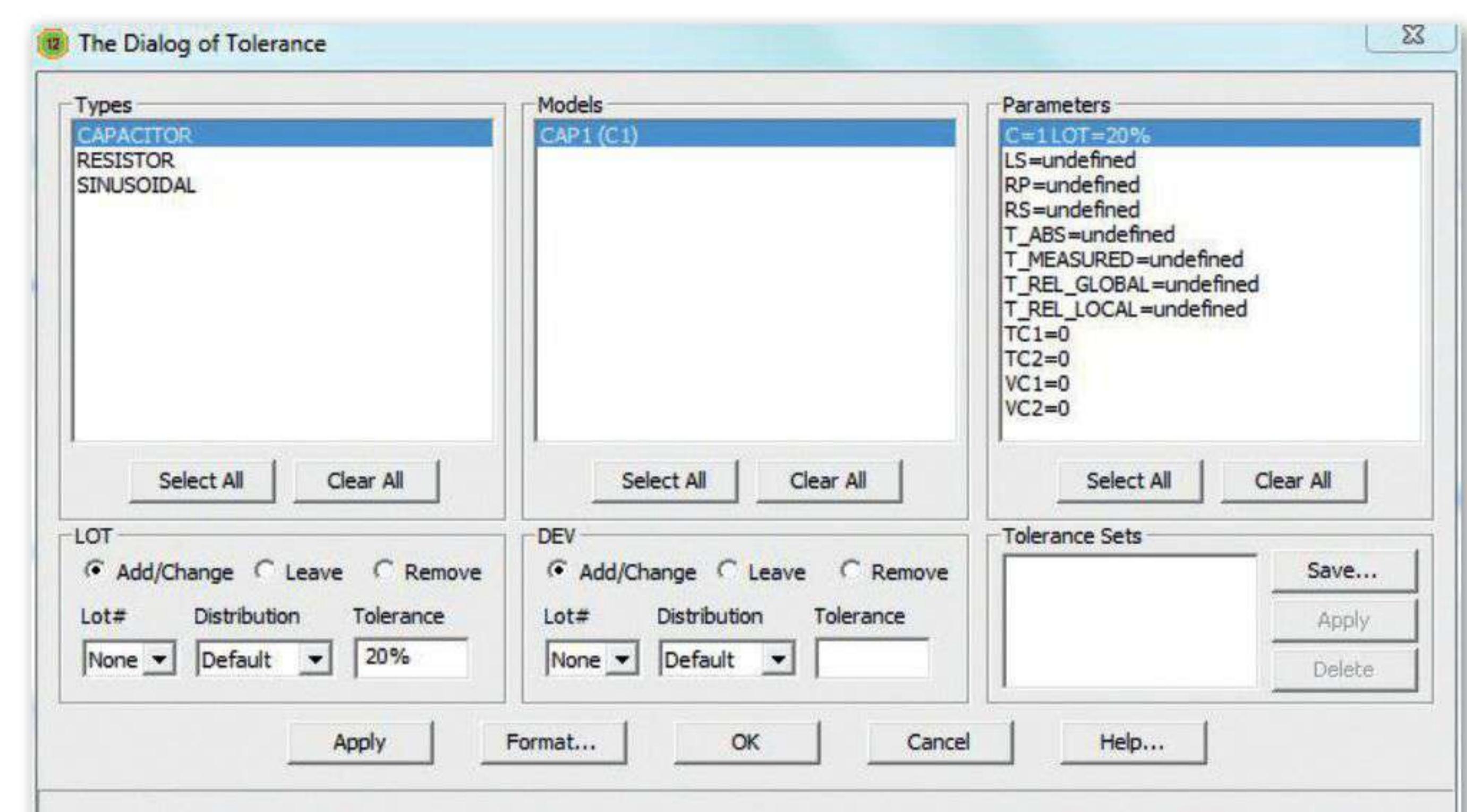


Bild 10: In diesem Dialog weisen wir dem Kondensator eine großzügige Toleranz zu ...

so aussehen wie in Bild 7. Variieren Sie nun statt des Kondensators den Widerstand zwischen 1k und 20k in 1k-Schritten und betrachten das Ergebnis, wiederum eine Kurvenschar. Falls Sie keine Kurvenschar sehen, prüfen Sie, ob Sie bei *Step It* ein Häkchen gesetzt haben. Fahren Sie mit der Maus über die Kurven, um den zu einer Kurve gehörenden Widerstandswert zu erfahren.

Das Stepping – also die vom Benutzer gezielte Variation der Werte einer oder mehrerer Bauelemente – steht in allen Analysen zur Verfügung. Bei einer Transientenanalyse könnten Sie zum Spaß die Frequenz der Sinusquelle V1 variieren und so eine interessante Sinuskurvenschar erzeugen und prüfen, an welchen Frequenzen sich Kurven schneiden.

Monte-Carlo

Bei der Monte-Carlo-Analyse werden mehrere Durchläufe durchgeführt. Für jeden Durchlauf wird eine neue Schaltung (circuit) erzeugt,

deren numerische Parameterwerte zufällig ausgewählt werden. Den Bauteilen zugeordnete Toleranzen werden dabei auf die Bauteilewerte angewandt. Modellparameter können relative DEV- und absolute LOT-Toleranzen besitzen. Symbolische Parameter wie Batterien, Spannungs- und Stromquellen können nur LOT-Toleranzen aufweisen. Toleranzen werden als Istwert oder als Prozentsatz des nominalen Parameterwertes angegeben. Eine LOT-Toleranz wird absolut auf jedes Gerät angewendet. Die Monte Carlo-Analyse ist in den Analysen Transient, AC, und DC verfügbar.

Führen Sie zunächst eine AC-Analyse ohne Stepping aus. Sollte Stepping aktiviert sein, ist im Limits-Fenster (Bild 4) der Button *Stepping* fettgedruckt zu sehen. Ist das der Fall, öffnen Sie den Stepping-Dialog und klicken dort auf *All Off*. Nach der AC-Analyse sehen Sie – wie oben beschrieben – die Graphen des Tiefpassfilters (Bild 5). Zur Monte-Carlo-Analyse müssen den Bauele-

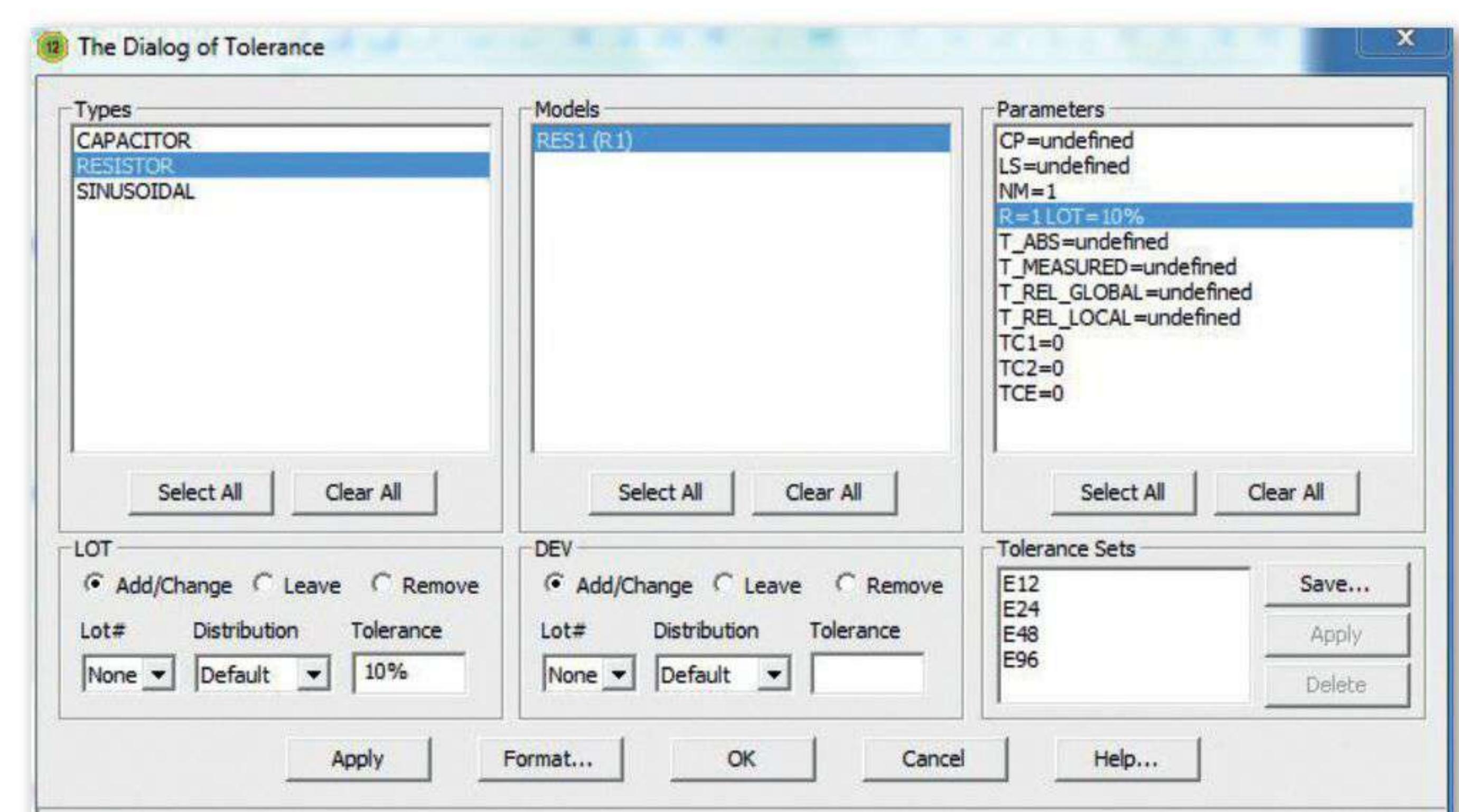


Bild 11: ... und die Toleranz für den Widerstand. So muss es unter Parameters aussehen.

menten eine Toleranz zugewiesen werden. Gehen Sie wie folgt vor: Im Menü *Monte Carlo* wählen Sie *Options*. Im aufpopgenden Dialog stellen Sie die Werte bitte so ein, wie in Bild 9 zu sehen ist. Die Verteilung (*Distribution*) ist auf *Worst Case* eingestellt, um maximale Toleranzen zu verwenden. Mit *Seed = 2* wird ein Zufallszahlengenerator gesetzt.

Rufen Sie über den Button *Tolerance* den Dialog zur Einstellung der Bauteiltoleranzen auf (Bild 10). In dieser Abbildung wurde die Toleranz des Kondensators bereits auf 20% gesetzt. Wählen Sie im Kasten *Types CAPACITOR*, unter *Models* (*C1*) und bei *Parameters* *C=1*. Falls Sie in einem Dialog nach einer Bezeichnung für das

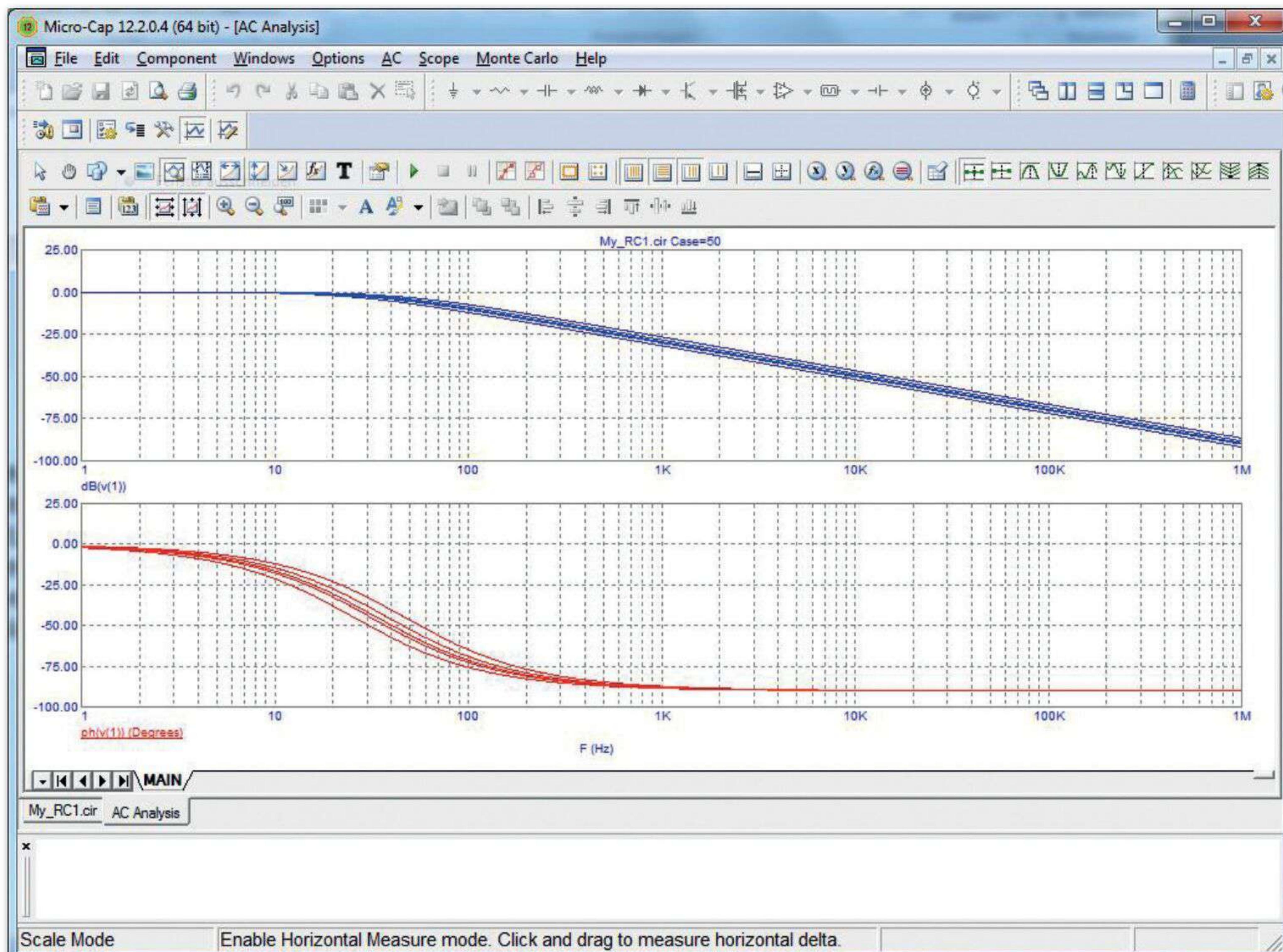


Bild 12: Das Resultat der Mühen: die Monte-Carlo-Analyse mit Extremwerten für den Worst-Case-Fall.

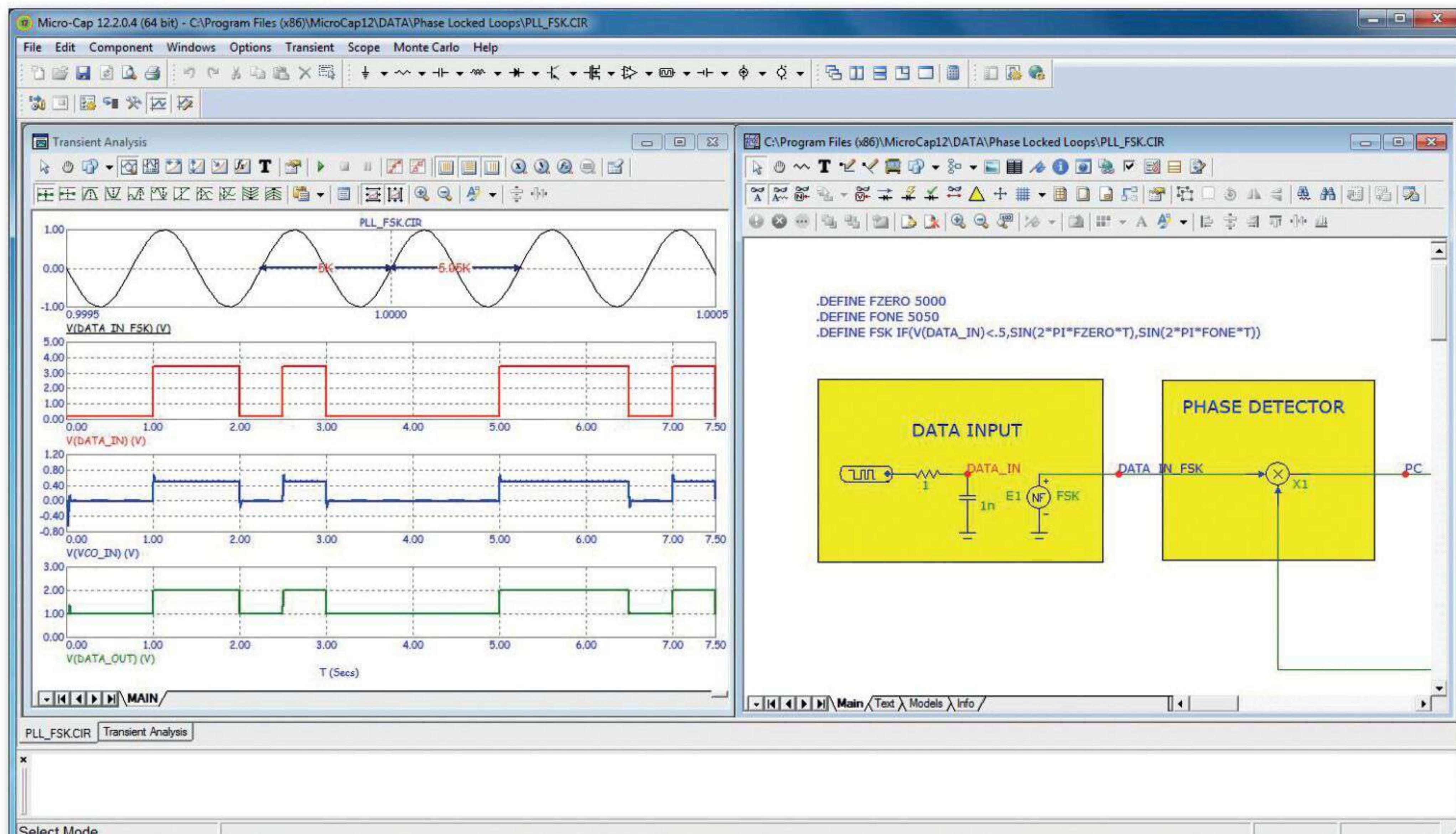


Bild 13: Viele Beispielanwendungen laden zum Lernen ein, wie hier ein Beispiel zur PLL.

Model zu C1 gefragt werden, vergeben Sie eine Bezeichnung wie z. B. CAP1. Nach der Auswahl in den drei Kästen wie in der Abbildung wird im Dialog der Bereich LOT veränderbar. Tragen Sie unter *Tolerance* 20% ein und betätigen den Apply-Button. Es sollte nun exakt so aussehen wie in Bild 10, das bedeutet, dass unter *Parameters* die blau markierte Zeile C=1 LOT=20% lautet.

Dasselbe führen Sie nun für den Widerstand R1 aus, siehe Bild 11. Anschließend sind die Toleranzen festgelegt und wir verlassen den Dialog mit OK. Stören Sie sich nicht daran, dass eine Toleranz von 20% für einen Widerstand unrealistisch ist, wir wollen doch eine hübsche Grafik erzeugen, deren Kurven man unterscheiden kann. Den noch offenen Dialog *Monte Carlo Options* können wir nun ebenfalls mit OK

verlassen. Starten Sie die Analyse erneut mit F2. Es sollten nun einige Kurven erscheinen, wie Bild 12 illustriert. Als Monte-Carlo-Verteilung haben wir Worst Case gewählt. Es stehen andere Verteilungen wie Uniform und Gauss zur Verfügung.

- Uniform: Es besteht eine gleiche Verteilung bei gleicher Wahrscheinlichkeit innerhalb der Toleranzgrenzen. Jeder Wert vom

Minimum bis zum Maximum ist gleich wahrscheinlich.

- Gauss: Wie der Name bereits ausdrückt, folgt die Wahrscheinlichkeit der Gaußschen Normalverteilung.

- Die Worst-Case-Verteilungen weisen eine 50%ige Wahrscheinlichkeit, das Minimum und eine 50%ige Wahrscheinlichkeit, das Maximum zu produzieren. Sie orientiert sich also an den Toleranzgrenzen.

Variieren Sie nun die Toleranzen der Monte-Carlo-Analyse und schauen, wie sie sich auf die Kurvenschar auswirken.

Zahlreiche Beispieldschaltungen (im Unterverzeichnis DATA) laden zum Lernen und Ausprobieren ein. Auf diese Beispiele beziehen sich die Erklärungen in den PDF-Dateien zum User Guide und der Referenz. Bild 13 zeigt beispielhaft eine PLL und deren Transienten-Analyse.

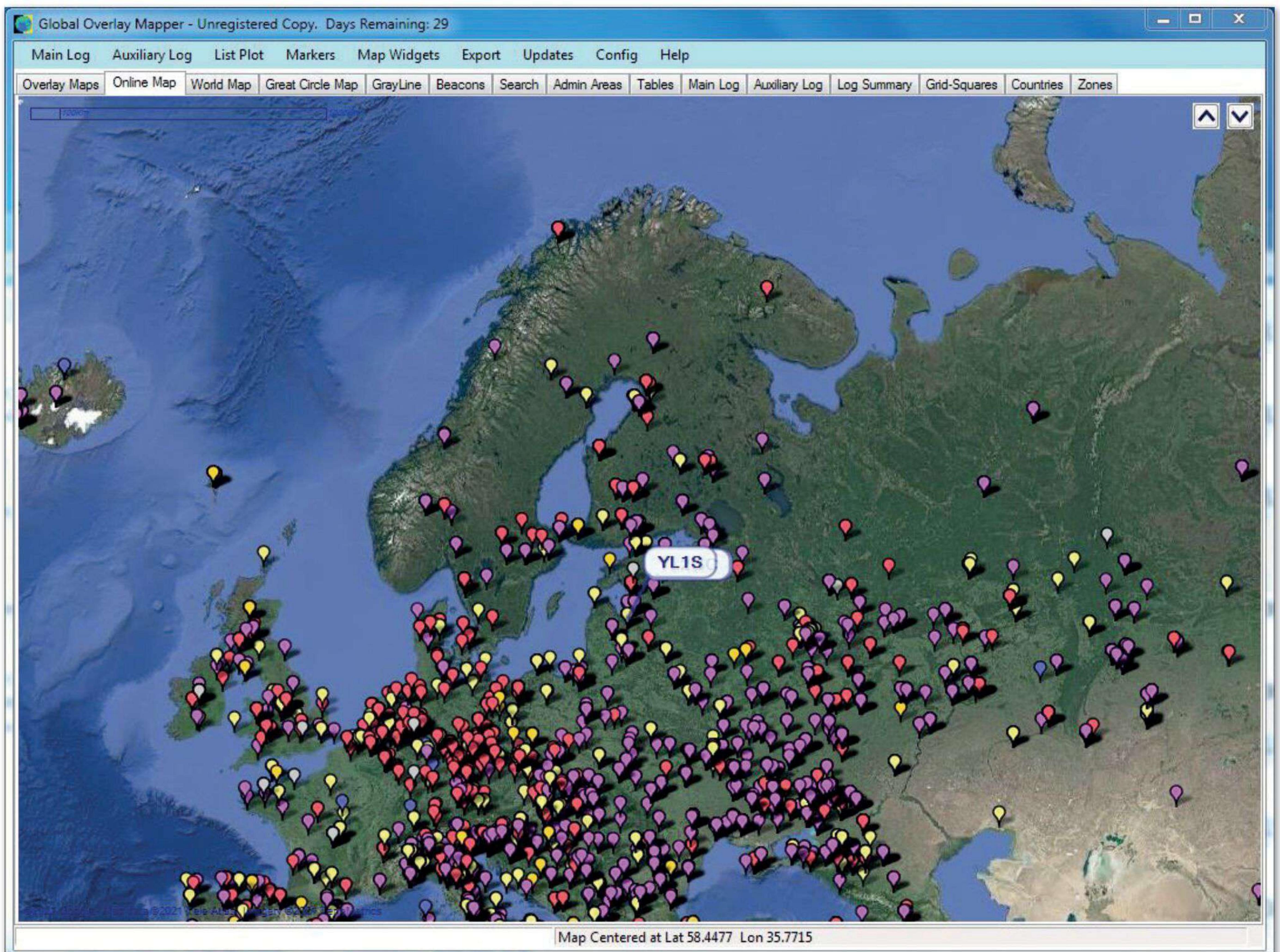
Abgesang

Hier steht kein Fazit, denn das wäre unangebracht, schließlich haben wir nur ein wenig an der Benutzeroberfläche gekratzt. Die Arbeit und das Lernen mit MicroCAP 12 hat dem Schreiber dieser Zeilen aber Spaß gemacht. Erstaunlich, dass man sich mehrere Tage jeweils mehrere Stunden mit der Analyse von nur drei Bauelementen beschäftigen kann! Die Leistungsvielfalt des Programms ist anfangs unüberschaubar. Je mehr man in die Tiefe vordringt, desto mehr Freude hat man mit dieser Software – und ein wenig mehr Durchblick. Diese fast 60 MB - auf der virtuellen DVD zum Download – lohnen allemal, um einige verregnete Winternachmittage nutzbringend zu gestalten.

Literatur/Verweise:

- [1] MicroCAP 12, Homepage: <http://www.spectrum-soft.com/index.shtml>
- [2] MicroCAP 12, Download: <http://www.spectrum-soft.com/download/download.shtml>
- [3] MicroCAP 12, Neues: <http://www.spectrum-soft.com/features.shtml>

Michael Wöste, DL1DMW



▲ Bild 1: Der Global Overlay Mapper (GOM) von Tim Makins, EI8IC.

Global Overlay Mapper: Ihre Logs auf der Karte

Tim Makins, EI8IC, ist ein Funkamateur mit Fernweh. Er bereiste alle Kontinente und die Zahl der von ihm besuchten Länder zählt über 150. Als freier Fotograf und insbesondere als professioneller Hersteller von Geo-Informationssystemen liegt es fast nahe, sein Kartenwissen für das schönste Hobby einzusetzen – es entstand die Software *Global Overlay Mapper (GOM)* für den Funkamateur. Nun ja, vielleicht hat er sein Programm ja auch ein wenig für sich selbst geschrieben. Man kann GOM

Mit den Jahren wächst das elektronische Logbuch. Wie sich Funkkontakte über den Globus verteilen (und mehr), das zeigt die Windows-Software Global Overlay Mapper. Die Redaktion hat die neue Version 5 per E-Mail angefordert und einen Monat getestet.

nicht ohne weiteres aus dem Internet laden, doch eine Mail an den Autor reicht aus und man erhält nach nur wenigen Stunden einen Download-Link zugesandt. Die Software – es handelt sich um eine 30 Tage mit kompletter Funktionsvielfalt ausgestattete Shareware – wurde von [1] geladen. Immerhin handelt es sich um stattliche 235 MByte, eine

Menge Bytes kommen sicherlich aufgrund der zahlreichen zum Paket gehörenden Karten zusammen. Eine flotte, stabile Verbindung zum Internet ist also förderlich. Aus rechtlichen Gründen durfte die Software leider nicht auf die virtuelle DVD. Ich habe das Programm unter Windows 7 getestet, obwohl es für Windows 10 autorisiert ist.

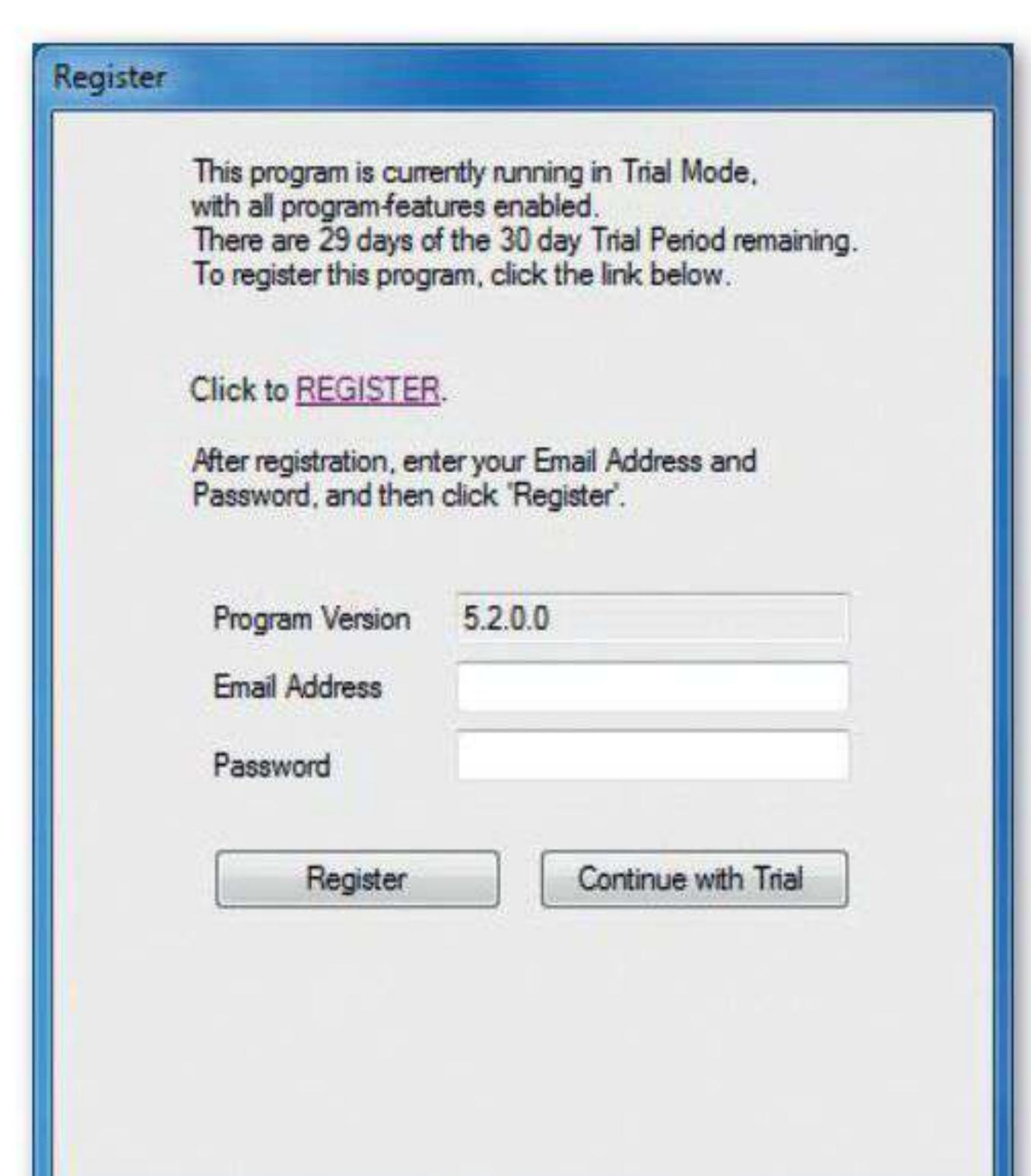


Bild 2: Nach dem Programmstart fragt GOM nach einer Registrierung oder weiter geht's mit der Testperiode.

Nach der Installation, die erfolgreich durchlief, konnte die Software gestartet werden (Bild 1). Damit man sofort loslegen kann, sind in der Datenbank etwa 2000 Logbucheinträge enthalten. Ohne



Bild 3: Die Overlay Map bietet mehrere Hintergrundkarten, darauf lassen sich verschiedene Ebenen einblenden.

weiteres zu testen, wurde eine eigene ADIF-Logbuchdatei mit 62904 Einträgen importiert und dabei die vorhandenen Logs überschrieben. Das dauerte 14 Minuten, etwa 4500 importierte Logbucheinträge je Minute. Neben ADIF importiert GOM auch das Cabrillo-Format. Je nachdem, welche Importoptionen man wählt, verlängert sich der Import um mehrere Minuten zur Erzeugung weiterer interner Tabellen. Bei meinem Versuch, die Logbuchdatei zu importieren kam es zur Fehlermeldung „Index out of Range“. Das Problem wurde dem Entwickler gemeldet und es entspann sich eine freundliche E-Mail-Kommunikation. Nach nur wenigen Tagen erhielt ich eine neue

Programmversion. Die Ursache lag jedoch nicht bei GOM, sondern in einem Fehler in einem Logbucheintrag. Es fehlte das wichtige MODE-Tag. Nach manueller Korrektur der Logbuchdatei wurde die riesige ADIF-Datei erfolgreich importiert.

Während des Imports versucht GOM, jedem Logbucheintrag einen Standort zuzuweisen, doch nicht immer ist der Ort oder der Maidenhead-Locator des Gesprächspartners notiert worden. Um dennoch zum Ziel zu gelangen, arbeitet GOM mit hamqth.com zusammen. Dazu muss der Benutzer dort angemeldet sein. Die Anmeldung ist kostenlos. In der Konfiguration bei GOM wird dann das Rufzeichen und Passwort eingetragen. Dieses als

Geocoding genannte Verfahren steht auch nach dem Import einer ADIF- oder Cabrillo-Datei im Menü *Main Log* zur Verfügung.

Hinweis: Bei jedem Start fragt GOM nach einer Registrierung (Bild 2), die etwas unter 24 Euro kostet und alle Leistungsmerkmale über die 30 Tage Testzeitraum hinaus freischaltet. Als Zahlungsart bietet der OM Paypal an. Das Programm in englischer Sprache verwaltet zwei Logbuch-Datenbanken, das *Main Log* (Haupt-Logbuch) und das *Auxiliary Log* (das alternative Logbuch). Die meisten Funktionen lassen sich – so schreibt der Entwickler – auf das Erstgenannte anwenden, ich habe keine Unterschiede feststellen können. Über Reiter (Tabs)

unterhalb der Menüzeile stehen verschiedene Karten und tabellarische Darstellungen zur Auswahl. Viele Funktionen aus den Menüs wirken sich auf die aktuell dargestellte Karte aus. Entscheidet man sich für die *Overlay Map* (Bild 3), wählt der Benutzer z.B. die Karte der ITU-Zonen, die Landkarte, welche die Landesgrenzen zeigt oder die hübsche Reliefkarte. Das sind die *background maps*, von denen immer nur eine aktiv sein kann. Über Schaltflächen lassen sich darin nun Großfelder, Längen- und Breitengrade, Länderbezeichnungen, Namen großer Städte, die Landes-Prefixe, IOTA-Nummern und Landesflaggen ein- und ausblenden – oder gar alles gleichzeitig auf „Ein“ setzen. Das geschieht program 技术isch betrachtet über sog. Ebenen (*layer*), die transparent „über die Karte gelegt“ werden – das sind die *foreground maps*. Vom intensiven Gebrauch dieser Ebenen hat GOM vermutlich seinen Namen abgeleitet: Global Overlay Mapper. Wie in Bild 3 dargestellt, lässt sich die Karte sehr einfach mit wenigen Klicks wunschgemäß optisch gestalten und je nach Anwendungsfall die eine oder andere Ebene ein- bzw. ausblenden.

Und wie kommen die Logs auf die Karte? Lassen wir sämtliche Logbucheinträge auf der Karte erscheinen! Im Menü *Main Log* wählen Sie *Calls* und dann *Plot all Calls*. Im Menü sehen Sie, es lassen sich zudem bestätigte und unbestätigte Logs anzeigen, QSO's nach Band oder nach Betriebsart filtern und als Tabelle oder auf der Karte mit geografischer Verteilung betrachten. Neben der *Overlay Map* stehen weitere Karten wie die *Online Map*, die *World Map* etc. zur Verfügung. Über kleine Pfeile rechts oben auf der Karte – wie bei der Online Map – kann man in sie herein- und

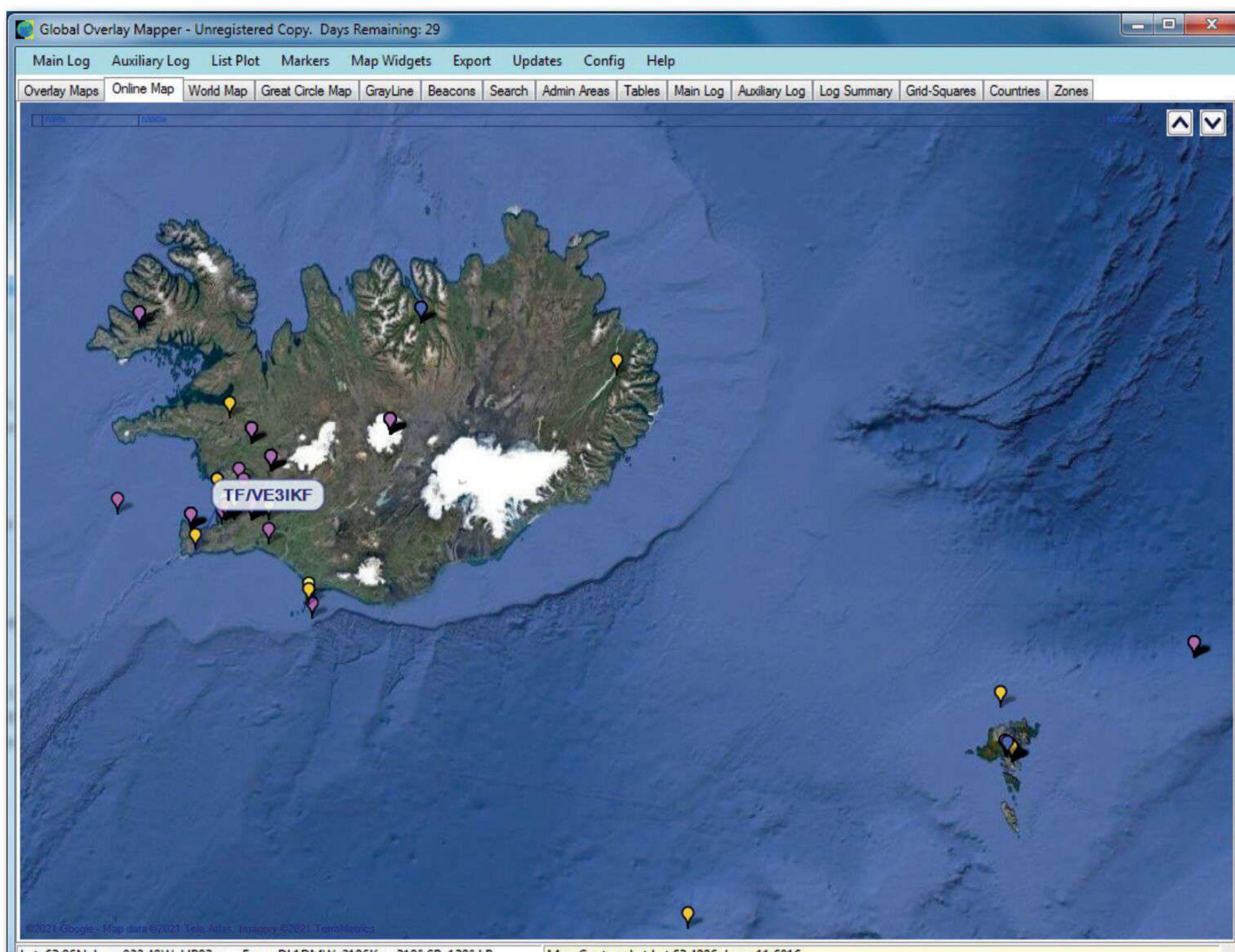


Bild 4: Die Online-Map zeigt bestätigte Logs auf der Insel. Der Vorteil der Online-Map ist die tiefe Zoomstufe.

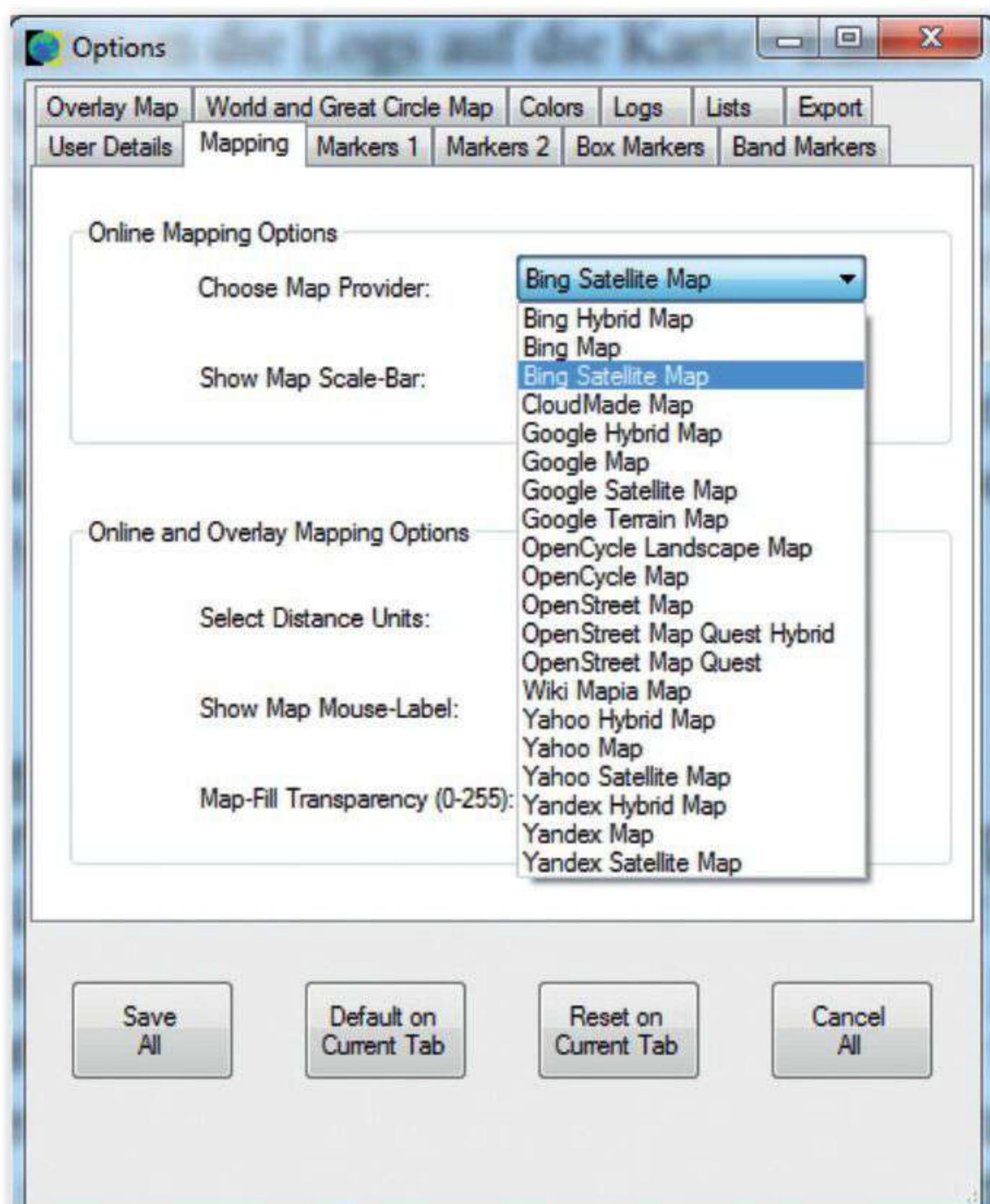


Bild 5: Über die Konfiguration kann man aus vielen Online-Karten auswählen.

herausfahren (zoomen) und damit eine detaillierte Darstellung gearbeiteter Stationen erhalten, wie in Bild 4 zu erkennen ist. Es handelt sich hier um die Darstellung aller bestätigten Stationen aus Island mit dem von mir bevorzugten Relief als Hintergrundkarte.

Die Online-Map

Diese Karte wird nicht auf dem PC lokal vorgehalten, sondern stammt von einem Kartenanbieter und wird über das Internet geladen. Der Vorteil: In diese Karte kann man sehr weit hinein zoomen und mit jeder Zoomstufe zeigt sie mehr Kartendetails. Das Verhalten ist von Google Maps oder Openstreetmap bekannt. Über die Konfiguration ist es möglich, aus einer Reihe von Providern wie Bing, Google, Openstreetmap, Yahoo und weiteren die gewünschte Online-Karte zu bestimmen (Bild 5). Der Entwickler weist allerdings darauf hin, dass nicht immer alle Karten funktionieren werden. Das liegt an dem Anbieter und die Art, wie er den Zugriff auf seine Karten gestaltet oder beschränkt. Dennoch kann man sich hier das heraussuchen, was optisch am besten gefällt.

Weltkarte

Die Weltkarte (world map) präsentiert im Mercatorprojektion alle Kontinente oder Teile daraus, wenn man in sie hinein zoomt. Sinnvoll

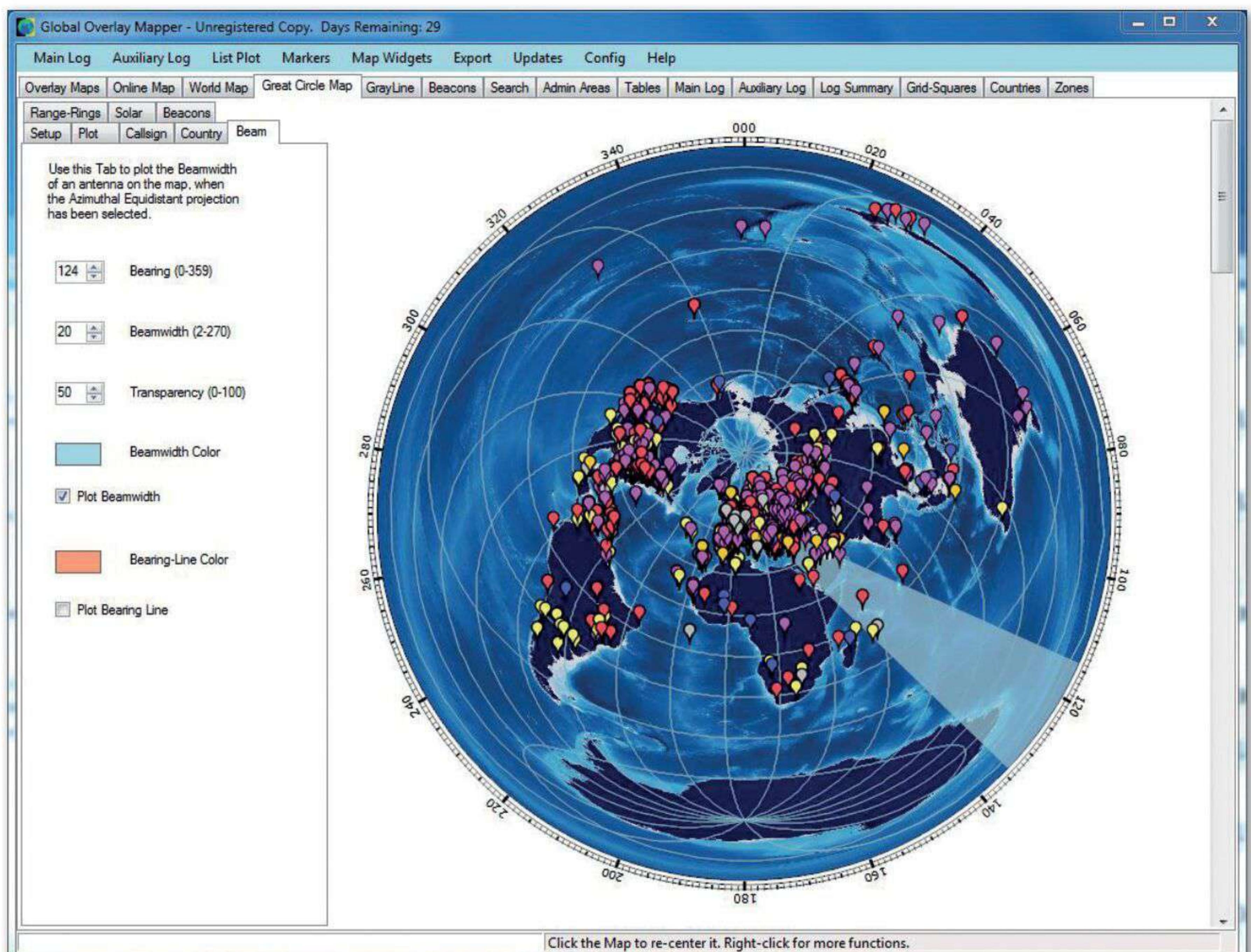


Bild 6: Großkreiskarte zur Bestimmung der Strahlrichtung. Die von der Antenne auf der Erde abgedeckte Fläche ist Richtung Südosten grau markiert.

wird die Weltkarte für den Funkamateuer durch hinzufügen verschiedener Ebenen. Im Wesentlichen sind es dieselben wie zuvor bei der Online-Map beschrieben: Das Einblenden von CQ-, ITU- und Zeitzonen, Grenzen der Kontinente und der ITU-Regionen, die Längen- und Breitengrade, das Maidenhead-Raster der Großfelder, Ländergrenzen und deren Prefixe, NCDXF-Baken, Hauptstädte der Länder und deren Bezeichnungen etc. macht aus der Weltkarte das, was sonst ein Globus oder die Weltkarte an der Wand leistet. Der Anwender wählt aus einer Fülle von sechzehn Hintergrundkarten die für ihn beste aus.

Great Circle Map

Die Großkreiskarte (Bild 6) dient dazu, die Antennenrichtung in Grad zu einer DX-Station zu ermitteln. Wurde die eigene Position des Heimatstandortes in die Konfiguration eingetragen, ist die Großkreiskarte nach dem Aufruf auf diese Position zentriert. Mit einem Mausklick in die Karte zentriert sie sich auf diese Position neu. Vier Projektionen stehen zur Auswahl, zwei beziehen sich auf die Sicht auf den Nord- und Südpol. Sendet eine NCDXF-Bake, zeigt die Großkreiskarte dies auf Wunsch an oder blendet wahlweise die Länder-Prefixe ein. Ein praktisches Feature ist die Anzeige der Beamabdeckung der Antenne (Reiter Beam). Nach Eingabe von

Öffnungswinkel und Beamrichtung markiert GOM die Fläche des Globus, die theoretisch von der Antenne abgedeckt wird (siehe Bild 6, der Beam zeigt nach Südosten). Gibt man im Reiter *Country* eine Landesbezeichnung vor, ändert sich die Beamrichtung auf der Karte entsprechend der Auswahl des Landes. Das funktioniert auch bei Eingabe eines Rufzeichens: Trägt man in den Reiter *Callsign* ein Rufzeichen ein und bestätigt mit der Enter-Taste, ermittelt GOM das Herkunftsland und richtet den Beam genau dorthin. Es wird noch besser: Nach Eingabe von *VK/ON5AB* erkennt die Software, das sich die Station nicht in Belgien, sondern in Australien befindet und richtet folglich den Beam in diese Richtung! Dazu erscheint in einem Infobereich allerlei Fakten über das entfernte Land sowie die Beamrichtung in Grad und die Entfernung in Kilometern.

Um auf der Karte Entfernungen besser abschätzen zu können, lässt man Ringe in definierter Entfernung um die eigenen Standort zeichnen. Das können auch mehrere Entfernungsringe sein, die GOM gestrichelt in die Karte plottet. Natürlich lassen sich in die Großkreiskarte die Längen- und Breitengrade einblenden. Und last, but not least, ist es möglich, die Großkreis-Hintergrundkarte aus zweiundzwanzig (!) optisch mehr oder weniger unterschiedlichen Karten zu wählen.

Gray Line

GOM wartet mit einer Gray-Line-Karte auf, nun ja, es ist mehr als nur eine Karte. Der Programmierer spricht richtig von einem Gray-Line-Werkzeug. Die aktuelle Dämmerungszone zeigt GOM, wenn man auf *Now* klickt und dann auf den Button mit der langen Bezeichnung *Gray Line from Date/Time Selector Value* klickt. Schon rutscht die Dämmerungszone an den aktuellen Standort.

Für jeden Ort auf diesem Globus lassen sich mit einem Mausklick die Zeit des Sonnenauf- und Untergangs berechnen.

Mehrere Arten von Animationen ermöglichen es, eine Vorstellung davon zu bekommen, wie die Dämmerungszone sich im Laufe der Zeit über mehrere Wochen und Monate hinweg wandelt, weil sich dabei auch der Winkel zwischen Sonne und Erde verändert.

Als Alternative zur Karte und als Möglichkeit, weit mehr Informationen zu erzeugen, als eine Karte zeigen kann, erzeugt GOM Tabellen nach Vorgabe einer aus elf vorgegebenen Aussagen bzw. Fragestellungen. Zuvor sind zwei Standorte – A und B – festzulegen, eine Bedingung / Frage auszuwählen und schließlich auf *Show Tables* zu klicken, um die Tabelle zu erzeugen. Folgende Aussagen bzw. Fragestellungen sind fix definiert:



Bild 7: Frei positionierbare Tafeln informieren den Benutzer über Call, Zeiten und Einträge aus dem DX-Cluster.

Bedingung 1: Standort A bei Sonnenaufgang und Standort B liegt im Dunkeln, Berechnung erfolgt für ein Jahr.

Bedingung 2: Standort A liegt im Sonnenuntergang und Standort B

ist im Dunkeln, Berechnung für ein Jahr.

Bedingung 3: Dämmerungszone sowohl für Standort A und Standort B, Berechnung für ein Jahr.

Bedingung 4: Dunkelheit für Ort A und Ort B, Berechnung für ein Jahr.

Frage 5: In welchen Ländern geht die Sonne auf oder unter, während es an Ort A dunkel ist? Berechnung gilt für das ausgewählte Datum.

Frage 6: In welchen Ländern ist es dunkel, während an Ort A die Sonne auf- oder untergeht? Berechnung erfolgt für das ausgewählte Datum.

Frage 7: In welchen Ländern geht die Sonne auf oder unter, während an Standort A am gewählten Datum die Sonne aufgeht?

Frage 8: In welchen Ländern geht die Sonne auf oder unter, während an Ort A die Sonne untergeht? Berechnung gilt für das ausgewählte Datum.

Bedingung 9: Berechnet die weltweiten Sonnenaufgangs-/Sonnenuntergangszeiten für alle Länder für einen Tag.

Bedingung 10: Erzeugt einen ganzjährigen Sonnenkalender für Ort A.

Bedingung 11: Ganzjähriger Sonnenkalender für Ort B.

Nach der Berechnung präsentiert GOM das Ergebnis im Reiter *Tables*. Nachfolgend ein Anwendungsbeispiel, das ich in dem Hilfetext des Programms fand: Nehmen Sie an, Sie wachen kurz vor der Morgendämmerung auf und fragen sich, in welchen Ländern gerade ein passender Gray-Line-Pfad existieren könnte. Um eine Antwort zu erhalten, stellen Sie Datum/Zeit auf *Now*, setzen Sie Ort A auf *Home Location* und wählen Sie **Frage 6:** In welchen Ländern ist es dunkel, während an Ort A die Sonne auf- oder untergeht? Klicken Sie auf *Show in Table*, es wird eine Liste der Länder mit UTC- und Ortszeiten für beide Enden des Pfades erstellt. Möchten Sie die Gray-Line-Bedingungen für den nächsten Funkwettbewerb wissen, setzen Sie das Datum des Contests ein.

Weltzeiten zur Verfügung und – wie praktisch – eine Tafel mit den Spots aus einem DX-Cluster.

Statistik nach Grid Square, Ländern, CQ- und ITU-Zonen

Der Reiter *Grid-Square* enthält eine Tabelle (Bild 8), die Auskunft darüber gibt, in welcher Betriebsart und auf welchem Amateurfunkband ein Maidenhead-Feld gearbeitet wurde. Im Kontextmenü einer belegten Zelle der Tabelle lässt sich ermitteln, wie oft dieses Feld auf allen oder nur auf diesem Band gearbeitet wurde. Im Kontextmenü findet man eine Funktion, die vom gewählten Maidenhead-Feld zur Karte führt und dieses Feld zur Anzeige bringt sowie weitere Funktionen.

Eine vergleichbare Tabelle erstellt GOM für Länder, CQ- und ITU-Zonen. Dem entsprechend bietet das Kontextmenü der Länder-, CQ-Zonen- oder ITU-Zonen-Tabelle die Möglichkeit zu erfahren, wie oft dieses Land bzw. die CQ-Zone oder diese ITU-Zone auf allen oder nur auf diesem Band gearbeitet wurde.

Fazit

Eine prima Software ist der Global Overlay Mapper für alle, die ihr Logbuch gewissenhaft führen und wissen möchten, wie sich Funkkontakte über die Erde verteilen, welche Länder oder Großfelder in der Sammlung noch fehlen und sicher auch für Statistikfans. Grafisch sehr ansprechend und mit vielen Kartenhintergründen ausgestattet, ersetzt es den Globus auf dem Schreibtisch und die Weltkarte an der Wand. Die Großkreiskarte hilft bei der Bestimmung der Beamrichtung und Gray-Line-Funker finden ein tolles Werkzeug zur Erhöhung der QSO-Wahrscheinlichkeit. Der 30-Tage-Testbetrieb und die Einarbeitung in GOM ist lohnenswert, danach muss jede(r) YL/OM für sich entscheiden.

Literatur/Verweise:

- [1] EI8IC-Homepage: mapability.com

Grid-Squares																
Grid Locator	160M	80M	60M	40M	30M	20M	17M	15M	12M	10M	6M	4M	2M	70CM	23CM	13CM
	Main Log	Auxiliary Log	Show Modes per Band				Show Bands per Mode									
JN29	0000100...	0000010...		0000110...		000011000000001000000000	00001...	0000010...				0000...	0000...			
JN33		0000010...		0000010...	0000000...	000011010000000010000000		0000010...		0000...						
JN34				0000110...	0000000...	0000100000000010000000				0000...						
JN35		0000000...		0000110...	0000000...	000001010000000000001000										
JN36	0000100...	0000000...		0000100...	0000000...	000001010000000000000000	00000...		0000...	0000...						
JN37	0000010...	0000000...		0000100...	0000000...	000001010000000000000000	00001...	0000100...			0000...	0000...				
JN38		0000000...	0000110...	0000100...	0000000...	000010000000000000000000			0000...		0000...	0000...				
JN39	0000100...	0000110...		0000100...		000010000000000000000000	00000100...	0000...	0000...	0000...	0000...					
JN40					0000000...	000010000000000000000000	00001...		0000...							
JN42		0000000...	0000100...			000010000000000000000000		00000100...		0000...						
JN44		0000000...	0000100...		0000110...	0000000...	000011010000000000000000	: N	0000100...	0000...						
JN45	0000100...	0000110...		0000110...	0000000...	000011010000000000000000	AM: N	CONTEST: N		0000100...	0000...					
JN46	0000100...	0000110...		0000100...	0000000...	000010000000000000000000	CW: N			00001...						
JN47	0000100...	0000110...		0000110...	0000000...	000010010000000000000000	FM: N			00000...	0000000...					
JN48	0000100...	0000110...		0000110...		000001000000000000000000	FSK44: N	HELL: N			0000...	0000...	0000...			
JN49	0000100...	0000110...		0000100...	0000000...		JT65: N	JT6M: N		0000100...		0000...	0000...			
JN52					0000000...	000010010000000000000000	JT9: N			0000100...						
JN54	0000000...	0000110...		0000110...	0000000...	000001010000000000000000	MFSK: N	MSK144: N		0000100...	0000...					
JN55		0000110...		0000110...	0000010...	000001010000000000000000	MT63: N			0000100...						
JN56		0000110...		0000100...	0000000...	000001000000000000000000	OLIVIA: N			0000...						
JN57	0000100...	0000110...		0000110...	0000000...	000000010000000000000000	PKT: N			0000...		0000...	0000...			
JN58	0000100...	0000110...		0000100...	0000000...	000001101000000000000000	PSK: N			0000...		0000...	0000...			
JN59	0000100...	0000110...		0000100...		000001000000000000000000	ROS: N	RTTY: N		0000...		0000...	0000...			
JN60						SSB: Y				0000100...						
JN62	0000110...			0000010...	0000000...	000001101000000000000000	SSTV: N			0000110...						
JN64	0000110...			0000110...		000001100000000000000000	THOR: N			00000100...						
JN66	0000000...	0000110...		0000100...	0000000...	000001100000000000000000	THRB: N			00000100...						
							TOR: N			00000100...						

Bild 8: GOM bietet weiteres wie diese Statistik Grid Square: Feld JN44 wurde auf 20m in SSB gearbeitet.

Michael Wöste, DL1DMW

Die Nummernsender Spionage- sendungen auf der Kurzwelle

Radio DARC brachte auf 6070 kHz in seinem wöchentlichen, einstündigen Programm einen Beitrag zu Nummernstationen. Das sind Sender, die auf der Kurzwelle Zahlenfolgen in AM oder SSB ausstrahl(t)en. Damit wurden bzw. werden Spione mit verschlüsselten Anweisungen versorgt.

Seitdem sind einige Monate vergangen und dennoch komme ich auf das Thema zurück und „lande“ – dank eines Links auf irgendeiner Webseite – auf der Internet-Präsenz der Zahlensender [1]. Diese Webseite hat allerlei Informationen zu Spionagesendern zusammengetragen. Die meisten Sender sind heute verstummt. Ich kann mich aber gut daran erinnern, dass ich als Jugendlicher oder als junger Erwachsener bei meinen Ausflügen in die Frequenzbereiche zwischen den Radiobändern oft unfreiwillig auf diese Zahlensender gestoßen bin – mit einem ganz normalen SSB-Kurzwellenradio. Wenn ich mich recht erinnere, war es damals ein Grundig Satellit oder zuvor der gute alte Yaesu FR-101. Soviel zur Nostalgie.

Stöbert man auf dieser Webseite, trifft man auf interessante Informationen: Ein STASI-Lexikon [2] oder da ist zum Beispiel „G06 Deutsche Lady“, ein Zahlensender, der – laut diesen Aussagen – heute noch aktiv ist. „Deutsche Lady“ ist ein Spitzname, den Hörer dem Sender gegeben haben, um Sender auseinander zu halten. „G06“ ist eine Nummerierung, das „G“ deutet auf die verwendete deutsche Sprache. Zur „Deutsche Lady“ steht dort zu lesen (Zitat):

„Seit 2014 ist G06 der einzige deutschsprachige Sender, der noch

aktiv ist. Verwendet die Funkvariante Deutsch für 2 und 5. G06 wird von einem Computer mit Windows 7 betrieben, der von der GRU betrieben wird. Wie E06 sendet auch dieser Sender wiederholt gefälschte Nachrichten zum Zwecke der Schulung, Verwirrung von Agenten usw. Diese gefälschten Meldungen werden über lange Zeiträume, bis zu einem Jahr, wiederverwendet.“

Die menschliche Neugier fordert es geradezu heraus, die Struktur der Aussendungen zu erkennen: Wann kommt ein Vorlauf? Wann werden Ziffern genannt, die darauf schließen lassen, dass sie einzelnen Spionen zuzuordnen sind? Wann beginnt der Nachrichtenblock für einen bestimmten Agenten, wie lang ist der und wann endet dieser. Den Nachrichteninhalt wird man

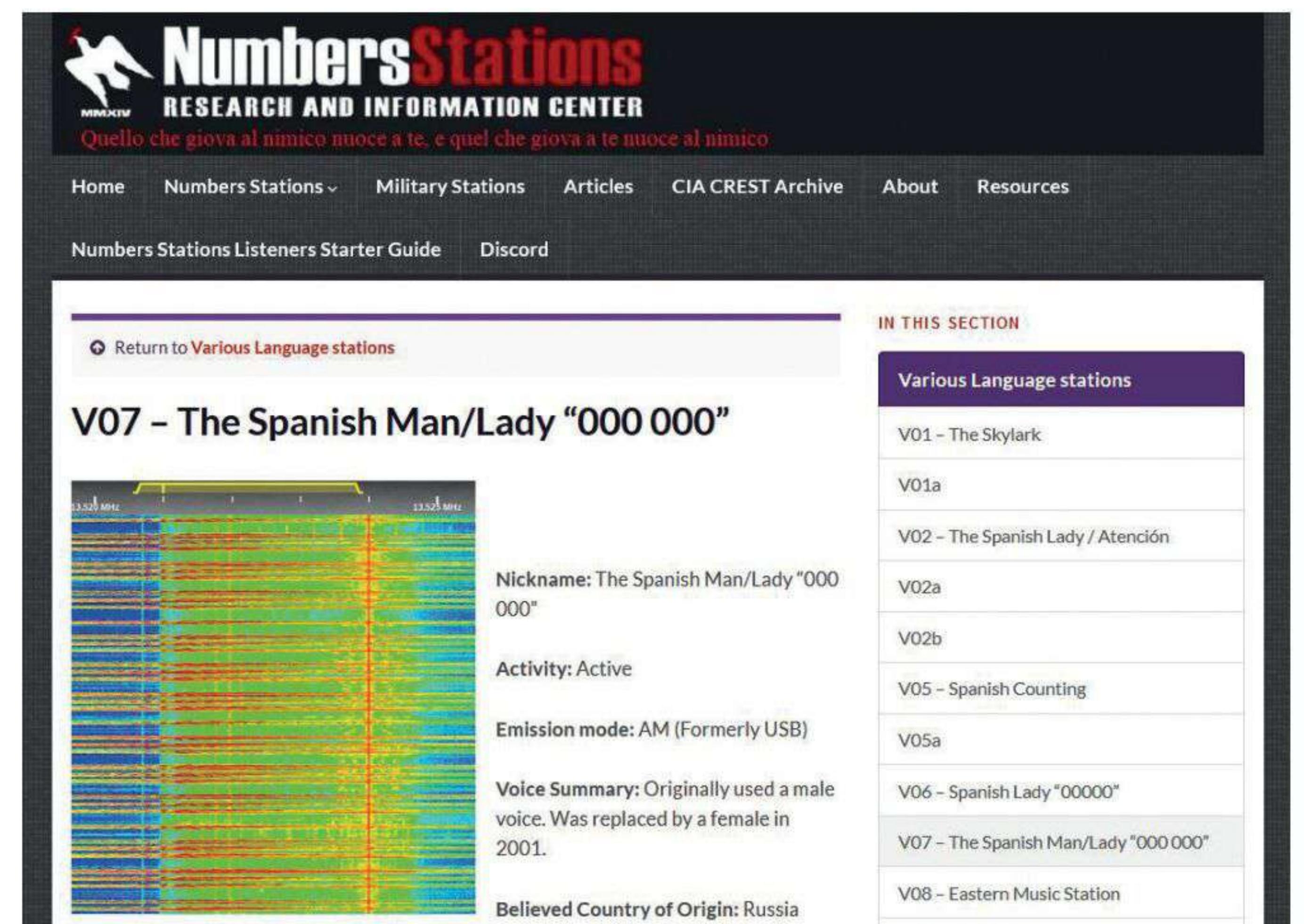


Bild 1: Informationen zu einem Zahlensender (mit Wasserfalldiagramm).

jedoch nicht entschlüsseln können. „G21 Musik und Morse“ ist ein Zahlensender, der inzwischen nicht mehr aktiv ist. Viel ist über ihn nicht bekannt, nur, dass zu Anfang der Sendung ein deutsches Volkslied gespielt wurde, einige Pfeiftöne zu hören waren, bis die Zahlenkolonnen monoton verlesen wurden. Auf der Webseite kann man sich einen Audiomitschnitt anhören, wie auch zu anderen Zahlensendern. Manche waren jedoch so kurz On Air, dass es keine Audioaufnahmen von ihnen gibt.

Wer heute in das Hobby „Zahlensender hören“ einsteigen möchte, findet einen „Numbers Stations Listeners Starter Guide“. Der Leitfaden enthält Informationen über die Verwendung eines SDR, den Kauf und Besitz eines eigenen SDR, Grundlagen des Signalempfangs und der Identifizierung sowie vieles mehr. Der Leitfaden listet wichtige Frequenzen von Militär- und Zahlensendern auf. Dabei ist die Bezeichnung „Zahlensender“ nicht

immer wörtlich zu nehmen, denn nicht immer werden Zahlenkolonnen verlesen. Wenn doch, geschieht dies heute meist automatisiert, entweder mit künstlicher Stimme oder einer realen Stimme, dessen Silben aufgezeichnet wurden.

Hier ist eine Station, die angeblich immer zu hören sein soll (Zitat): The Air Horn ist eine russische Militärstation, die im westlichen Militärbezirk betrieben wird. Er sendet auf 3510,00 kHz (Amateurfunkband!) und strahlt seine Kanalmarkierung rund um die Uhr aus. Ursprünglich funkte der Sender auf der Frequenz 4020,00 kHz. Diese Station sendet keinen Nachrichtenverkehr, stattdessen wird sie zum Testen von technischen Geräten und anderen Audiotests verwendet. Wir haben festgestellt, dass sie dazu neigt, an zufälligen Tagen nicht mehr zu senden, um dann ein paar Tage später wieder aktiv zu werden. (Zitat Ende). Gilt das noch?

Priyom

Durch die Lektüre Volkers Webseite (SM5ZBS) [3], der sich bereits 2014 dem Thema zugewandt hatte und darüber berichtete, wurde ich zufällig auf Priyom [4] aufmerksam. Priyom ist eine internationale Organisation, die es sich zur Aufgabe gemacht hat, die geheimnisvolle Realität der nachrichtendienstlichen, militärischen und diplomatischen Kommunikation über Kurzwellenradio zu erforschen und ans Licht zu bringen. Priyom schreibt über sich (aus dem englischen übersetztes Zitat): „Priyom.org

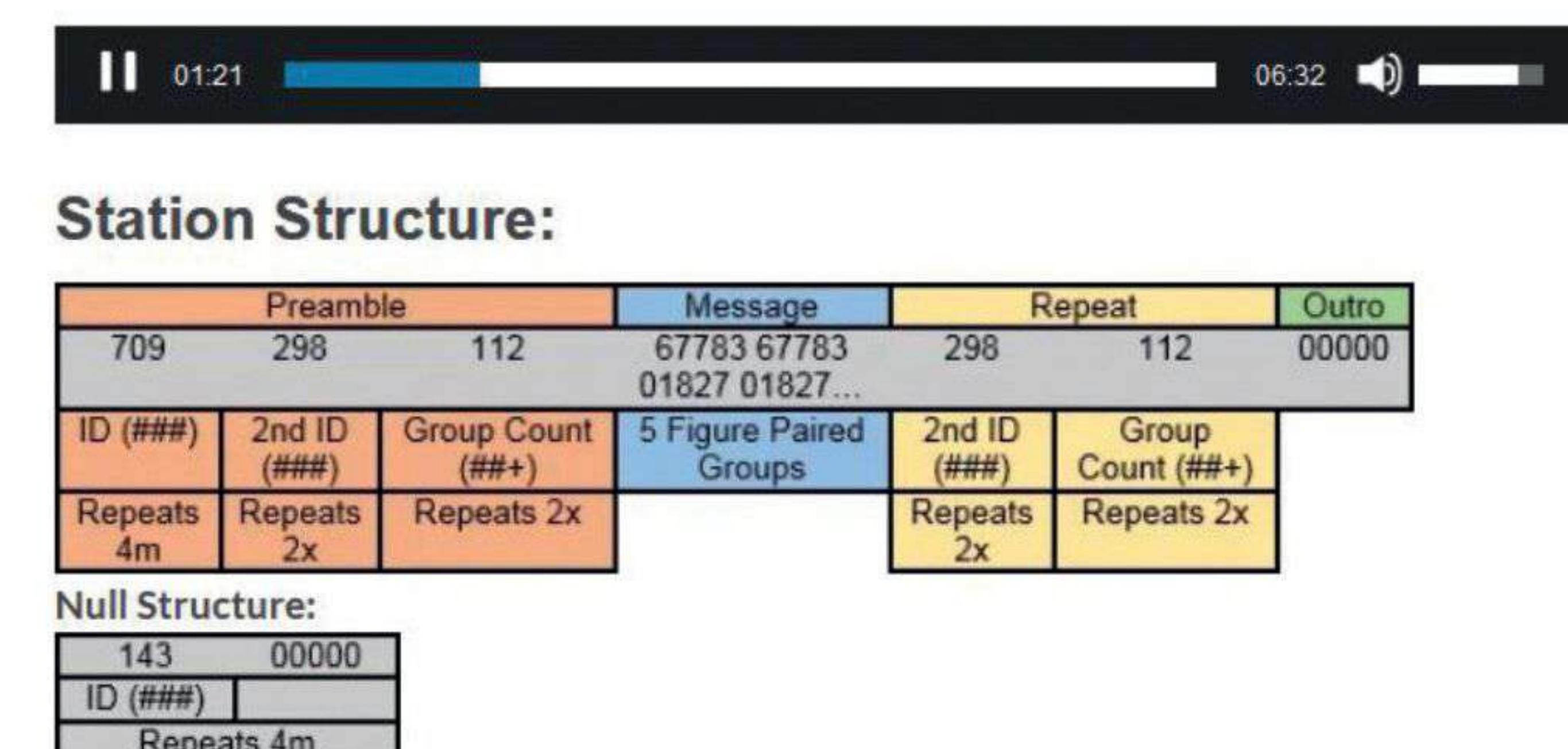


Bild 2: Hörer analysierten die Struktur einiger Nummernsender.
Quelle: [1].

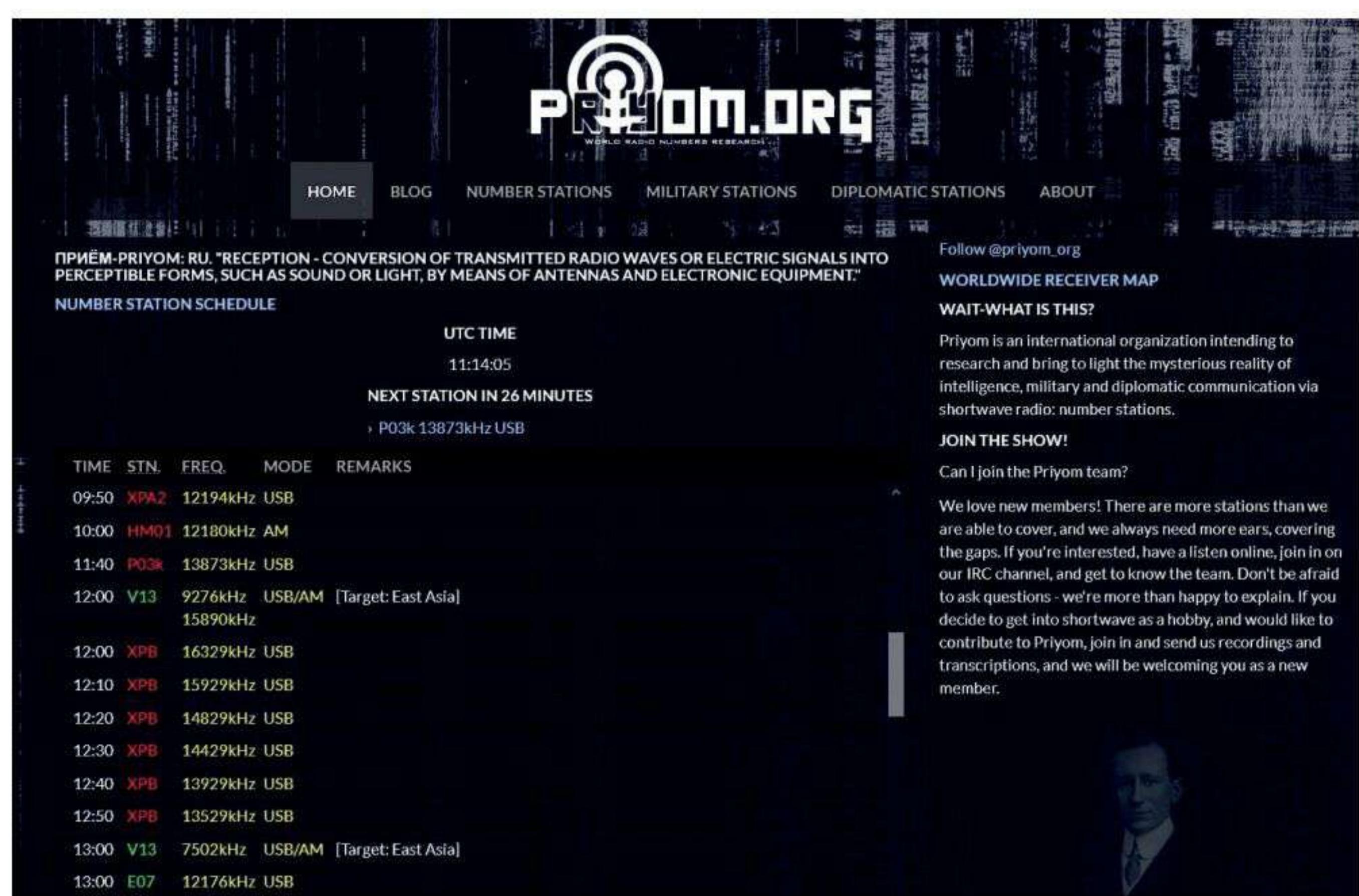


Bild 3: Diese Infoseite erleichtert den Einstieg in den Empfang von Zahlensendern durch Zeit- und Frequenzangabe.



Bild 4: Zu allerlei Spionagesendern hält Priyom.org teils Wasserfall-diagramme und Audiomitschnitte bereit.

ist eine internationale Gruppe von Radio-Enthusiasten, die sich im Jahr 2010 zusammenfand, als die Aktivität der als UVB-76 bekannten Station zunahm. Das Mysterium dieser Station inspirierte uns dazu, nach anderen ähnlich mysteriösen Stationen zu suchen, und damit in die bizarre Welt der berüchtigten Nummernstationen des Kurzwellenradios einzudringen. Stark inspiriert von der Arbeit anderer Online-Gruppen, wurde Priyom entwickelt, um sowohl die aktuellen als auch die historischen Informationen über Nummer-Stationen auf einfache Weise zu präsentieren und die Aufnahme, das Abhören und die Analyse von Kurzwellenradio, Nummer-Stationen und allen damit verbundenen On-Air-Rätseln allen Interessierten zugänglich zu machen.“ (Zitat Ende).

Auf der Startseite liefert Priyom eine Empfangshilfe für Soforteinstieger: Eine Tabelle zeigt die (aus Erfahrung vermuteten) nächsten Sendungen von Zahlensendern, deren Startzeit und Frequenz. Ein erster Test brachte zur angegebenen Uhrzeit zwar keine hörbaren Zahlenkolonnen zutage, es war jedoch das für einige

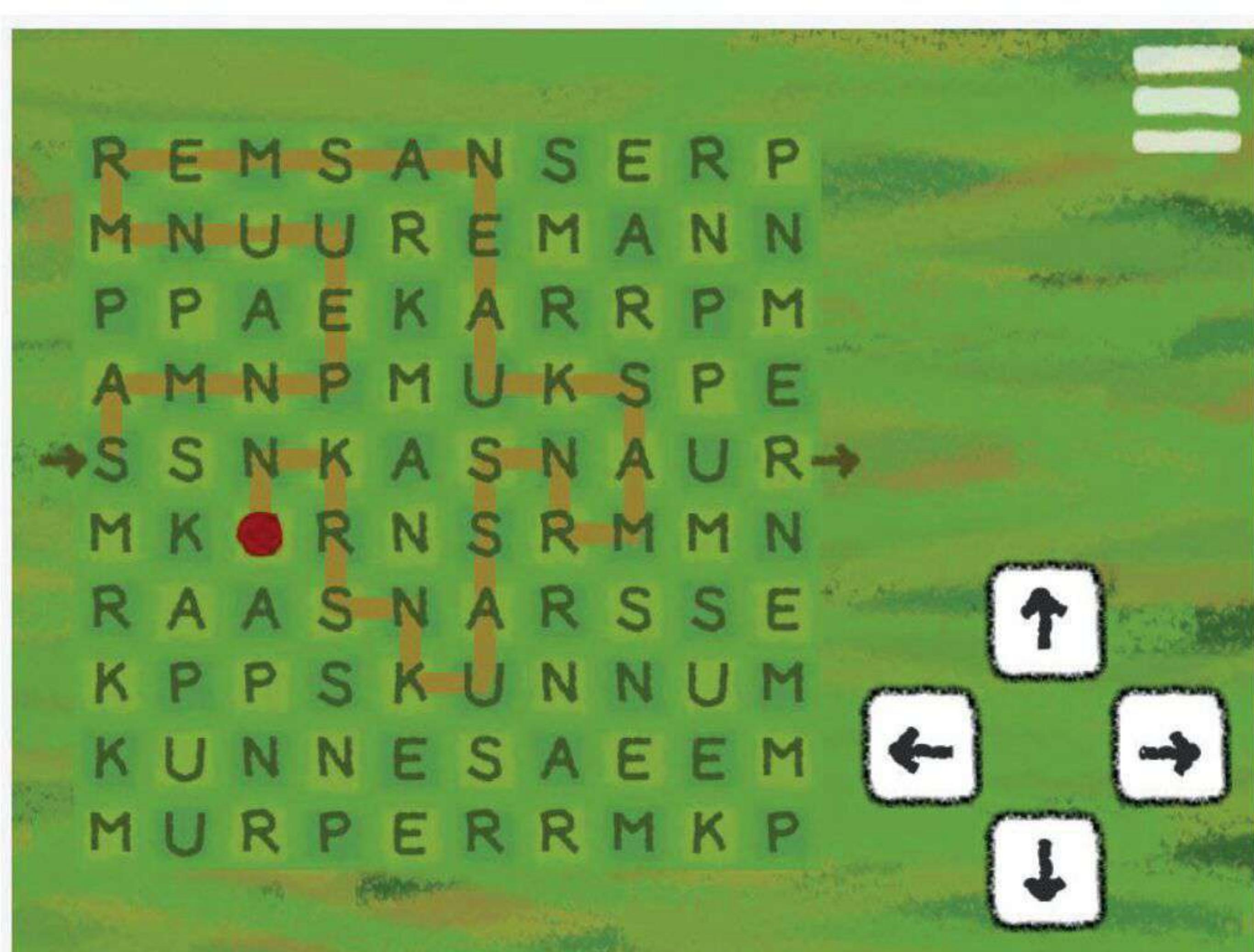
Sender charakteristische Ticken einer Uhr zu hören, oft ein Zeichen, dass bald eine Aussendung beginnen wird. Ein anderer Empfangsversuch auf 14,829 MHz um 12.20 UTC verlief erfolgreicher: Pünktlich startete der Sender, übertrug jedoch keine gesprochenen Zahlenkolonnen, sondern ein 16-Ton-MFSK-Signal. Laut Priyom handelt es sich um den Sender XPB aus Moskau. Die Sendung dauerte etwa 1 Minute. Priyom bietet zu etlichen Sendern eine Beschreibung und – wie in diesem Fall – ein Audiomitschnit und ein Wasserfalldiagramm. Priyom's Webseite ist ein guter Ort zum Start in den Empfang von Zahlensendern.

Literatur/Verweise:

- [1] Number Stations: <https://www.numbers-stations.com/>
- [2] STASI-Lexikon: <https://www.numbers-stations.com/articles/stasi-germanrussian-lexicon-of-intelligence-terms/>
- [3] Volkers Webseite, Zahlensender: <https://elektronikbasteln.pl7.de/zahlensender-auf-kurzwelle-abhoeren>
- [4] Priyom.org: <https://priyom.org/>

Michael Wöste, DL1DMW

Lernspaß garantiert: Das Morsespiel



Auf dem Schreibtisch finde ich einen Zettel mit der Internet-Adresse tetopia.de. Ich hatte sie mir irgendwann einmal notiert. Neugierig tippe ich die Adresse in den Browser ein und gelange zum Morsespiel von Theresa Thoma, DC1TH.

Bild 1: Der Spieler durchschreitet den Moorsumpf. Ertönende Morsezeichen führen durch den gefahrlosen Weg.

Morsen zu lernen ist eine Fleißaufgabe und benötigt beständiges Durchhaltevermögen. Hat man ungefähr zehn Buchstaben erlernt, mag das Lernprogramm als ein wenig langweilig empfunden werden. Mit einem Morsespiel kann es genauso gehen: Die Abläufe wiederholen sich schnell – die Motivation lässt nach. Nicht so bei Theresas Morsespiel.

Das ist anders: Dieses Morsespiel besteht nicht aus einem Spiel, sondern aus mehreren, unterschiedlichen Spielen. Ist man eines Leid, wählt man ein anderes und übt weiter. Man spielt es auf einem Android-Handy, auf einem Tablet (Android), dem PC unter Windows oder Linux. Hier spielen wir mit der Demoversion auf Theresas Webseite [1]. Es ist die Onlineversion des Spiels: Nichts muss auf dem PC installiert werden – man surft die Webseite an und schon geht's los. Dieser Bericht fußt auf die Erfahrungen mit eben dieser Onlineversion.



Bild 2: Das Menü der Onlineversion mit Spielauswahl und Einstellungen.

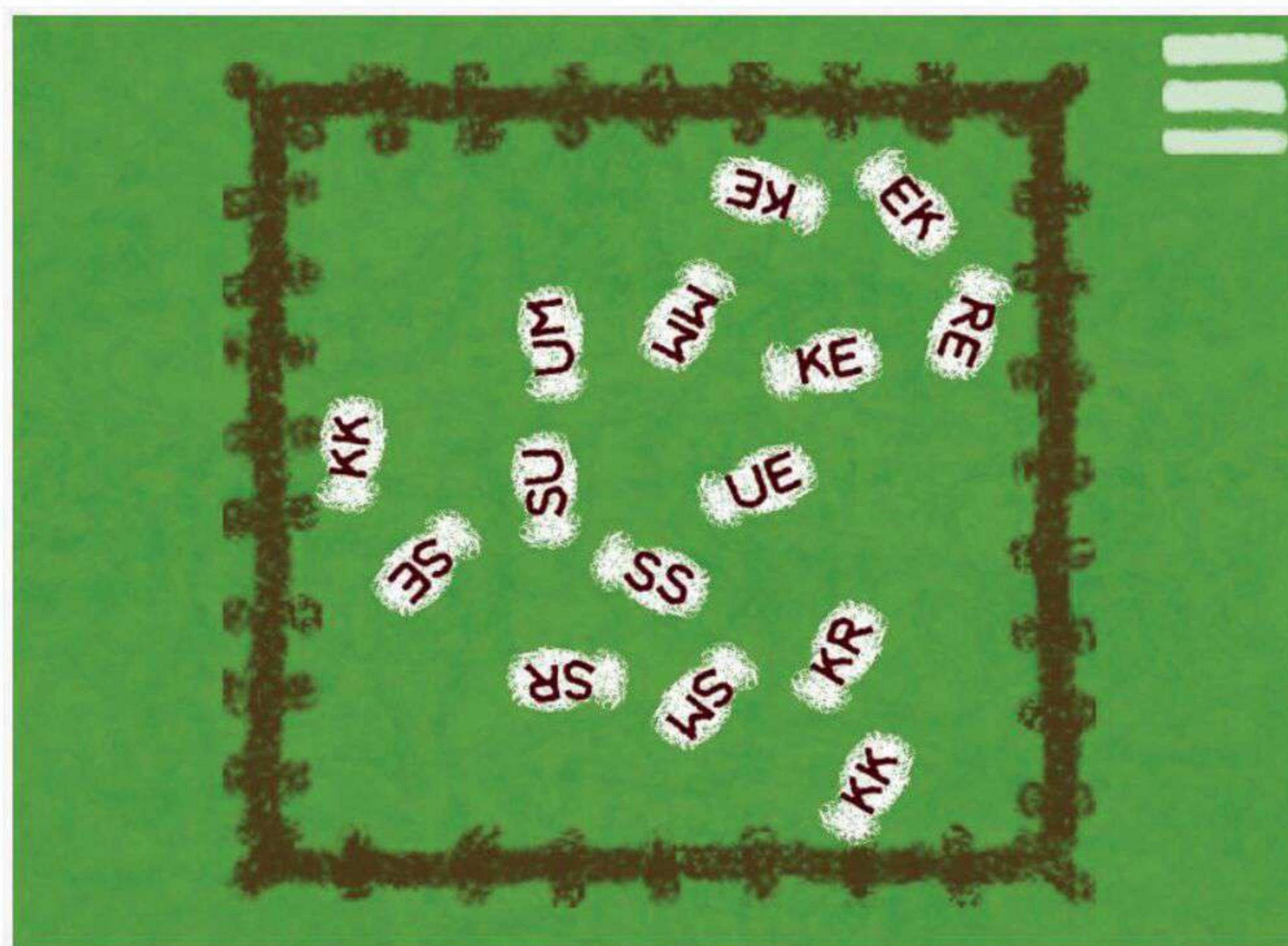


Bild 4: Die Schafe melden sich mit ihrer Morsekennung, der Spieler klickt sie von der Bildfläche und gewinnt, wenn alle Tiere „gefangen“ wurden.

Das Aufmacherfoto zeigt mein Lieblingsspiel, den Moorsumpf: Der Spieler muss, ohne zu versinken, das Moor durchschreiten. Dazu gibt der PC ein Morsezeichen vor und man muss nun mit den Pfeiltasten der Tastatur den erkannten Buchstaben anfahren. Wenn alles gut läuft und alle Buchstaben richtig erkannt sind, gelangt man unversehrt über einen verschlungenen Pfad auf die andere Seite des Moores und hat – gewonnen! Doch schon ein falsch erkannter Buchstabe führt vom sicheren Weg ab und man versinkt im Moor – blubb, weg ist der Spieler! Macht nichts, ein neuer Versuch ist garantiert.

Bild 2 zeigt das Menü des Onlinespiels und die Einstellungen. Hier justiert der Spieler die Geben- und Hörgeschwindigkeit, bestimmt

die Zahl der Buchstaben zum Hören und Geben nach der Koch-Methode oder wählt ein Spiel aus. Eher klassisch gehalten ist das Buchstabenraten (Bild 3). Der PC piepst den Buchstaben als Morsezeichen und der Spieler klickt mit der Maus auf den erkannten Buchstaben. Eine schöne Idee ist das Schafefangen-Spiel (Bild 4): In einem umzäunten Bereich weiden Schafe mit individueller Kennung. Statt des üblichen „Mäh“ melden sie sich natürlich in der Morsesprache und Blöken die eigene Kennung. Wird diese vom Spieler richtig gehört und das betreffende Schaf mit der Maus angeklickt, hört man eine akustische Belohnung und es verschwindet von der Bildfläche – es ist „gefangen“. Zögert der Spieler einen Moment mit der Antwort, ertönen

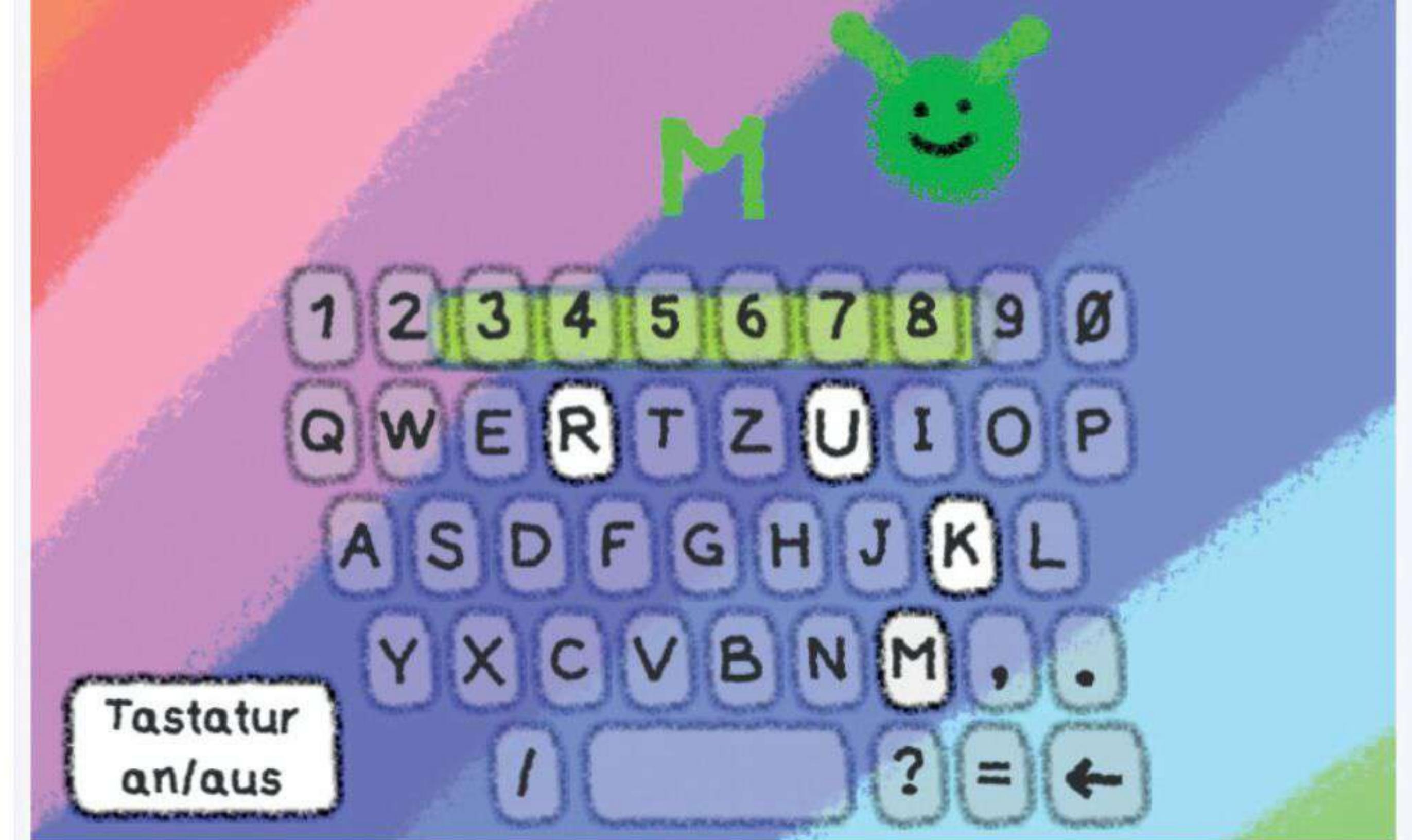


Bild 3: Klassisch: Der PC morst ein Zeichen, der Spieler klickt den erkannten Buchstaben auf der Tastatur an.

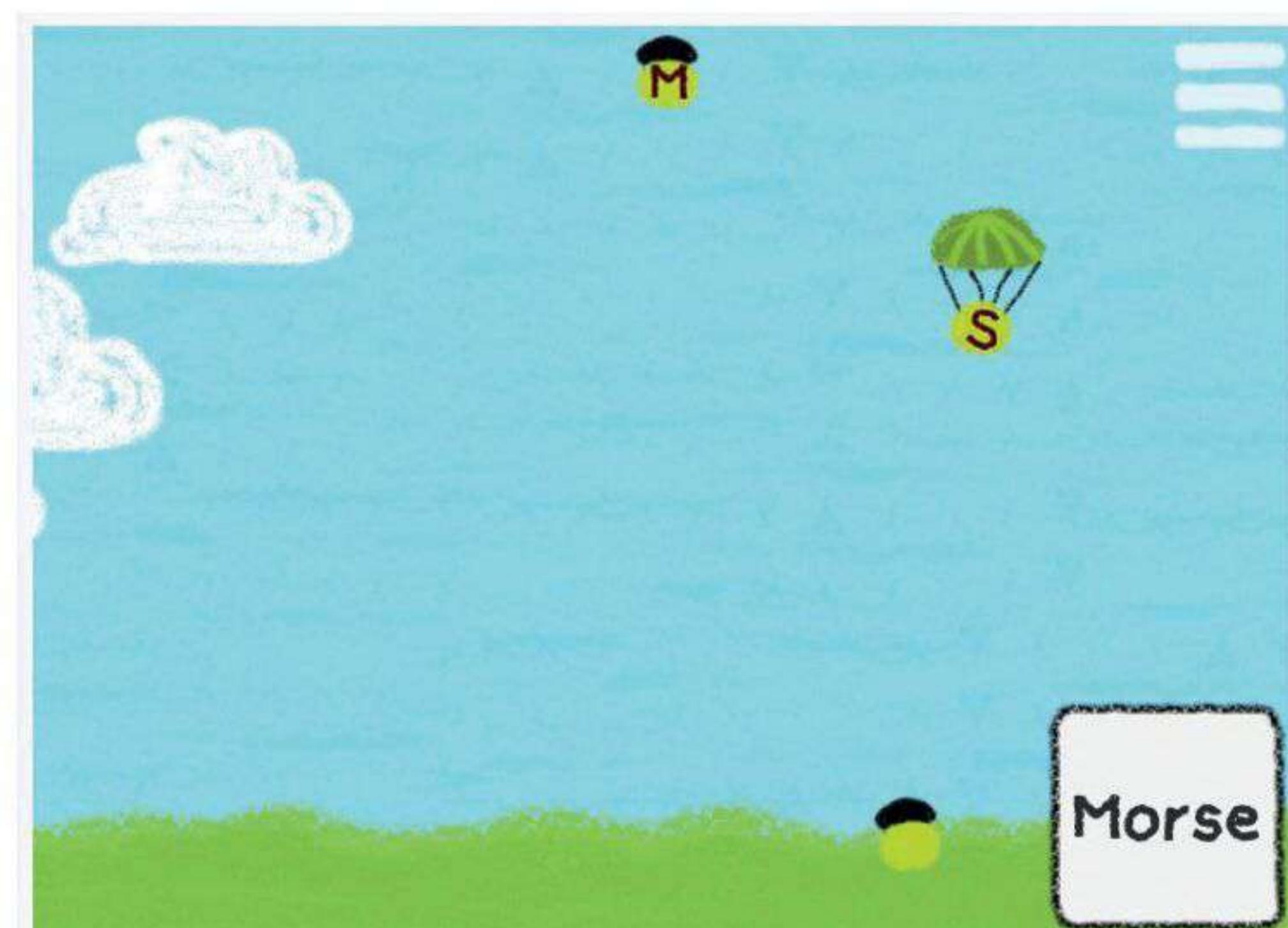


Bild 5: Fallschirme öffnen sich bei richtig gehörtem Morsecode und „retten“ den Springer.

die Morsezeichen erneut. Wurden schließlich alle Schafe gefangen, hat der Spieler gewonnen.

Ein Spiel mit Gebeübung ist das Fallschirmspiel: Vom Himmel fallen nacheinander gemächlich Fallschirmspringer mit Buchstabenkennung. Mittels Leertaste der Tastatur oder mit der Maustaste morst der Benutzer diesen Buchstaben und „rettet“ damit den Springer, weil sich bei erkannter Morsekennung der Fallschirm öffnet (Bild 5) und der Springer langsam zu Boden segelt. Wie gut man Buchstaben gehört hat, verrät die Statistik (Bild 6). Weitere Spiele sind in der Demo-version entweder nicht vorhanden oder werden erst nach Erreichen eines Leistungslevels zugänglich. Ich habe es nicht darauf angelegt, diese Stufen zu erreichen, daher bleiben mir diese verborgen. Dennoch erfuhren meine verschütteten Morsekenntnisse beim angeregten Spiel einiges „Aufpolieren“.

In der zum Download kostenlosen Vollversion sind einige nette Spiele erst nach Erreichen einer Leistungsstufe frei geschaltet, wie das Video auf YouTube.com [2] verrät. Nicht gesehen habe ich das Spiel Tunnel oder Schiffe versenken. Davon erfährt man im Video: Theresa, DC1TH, stellte im Rahmen eines Kolloquiums im Jahre 2019 ihr Spiel vor. Der DARC schnitt den Vortrag mit und stellte ihn bei YouTube ein. Fazit: Sehens- und spielerisch, nicht nur für Jugendliche, aber auch für diese Gruppe! Morsespaß ist garantiert.

Literatur/Verweise:

- [1] Homepage: https://www.tetopia.de/programmieren/morse_demo.php
- [2] Vortragsvideo des DARC zum Morsespiel: <https://www.youtube.com/watch?v=WT9AQo16fMQ&t=631s>

Michael Wöste, DL1DMW

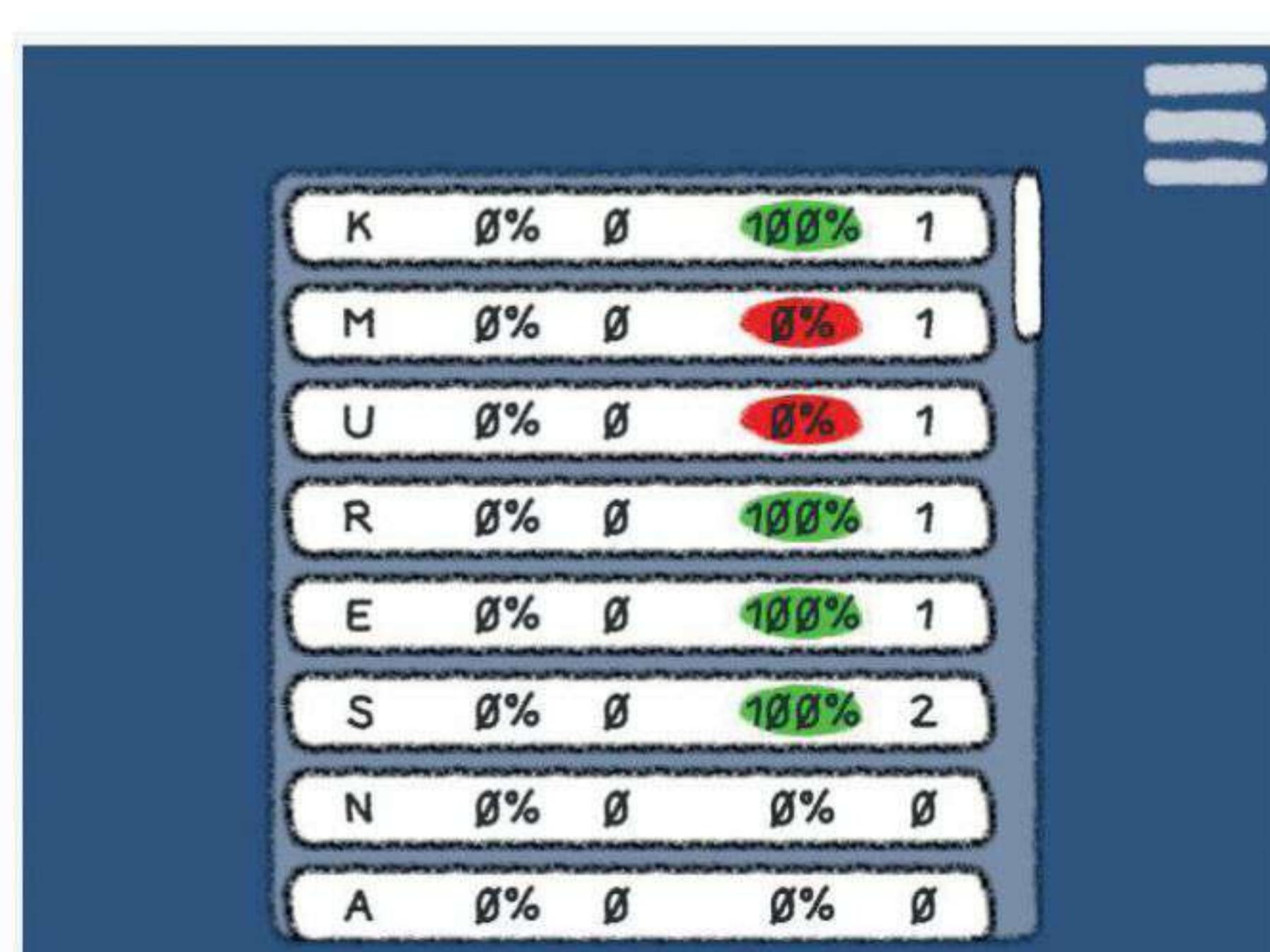


Bild 6: Die Statistik gibt für jedes Zeichen Auskunft über den Leistungsstand. Hier der Beweis des ungenügenden Hörfähigkeiten des Autors. Buchstaben mit schwachen Leistungsdaten werden öfter wiederholt.

Bonus- Lesestoff:

Das Buch im Format PDF auf der Heft-DVD – inzwischen auf der virtuellen DVD zum Download – folgt einer inzwischen lieb gewordenen Tradition: Jede DVD hält für die Leserin, den Leser ein für Funkamateure interessantes Buch kostenfrei bereit: War es im letzten Jahr das tolle Magnetantennenbuch, geht es heute um die beliebten KW-Drahtantennen für Funkamateure.

Kurzwellen Drahtantennen für Funkamateure

Alfred Klüß, DF2BC



Das Kurzwellen-Antennebuch von Alfred Klüß, DF2BC.

„Kurzwellen-Drahtantennen“ als PDF-Buch

Das Buch des Autors Alfred Klüß, DF2BC, führt auf über 150 Seiten in das Thema ein und löst unter anderem das Problem, falls am Ende des Grundstücks noch eine Menge Drahtantenne übrig bleibt. Das Werk behandelt Indoor- und Outdoor-Drahtantennen sowohl für den Sendeamateuer, als auch für den Kurzwellenhörer.

Das Buch startet mit praktischem Antennenbau: Welche Materialien eignen sich, wie wird eine Drahtantenne aufgehängt – welche Mittel- und Endisolatoren sind geeignet, wertvolle Tipps sowie ein konkreter Bauvorschlag erleichtert dem Leser den Aufbau. Der Text widmet sich nun dem Thema Einspeisung und dem Wickeln von Baluns. Unter Einhaltung dort verratener Praxistipps entstehen professionell aussehende und wasserdichte Einspeisungen, Zugentlastungen und Übergänge zwischen Koaxialkabel und „Hühnerleiter“. Zahlreiche Tipps und Tricks gehen detailliert

auf mögliche Fallstricke und deren Vermeidung ein.

Ist die Antenne aufgehängt, kommt der Abgleich. DF2BC beschreibt diesen Arbeitsschritt, stellt notwendige Messgeräte vor und kommt anschließend auf das wichtige Thema Erdung zu sprechen. Nach diesen eher für alle Drahtantennen geeigneten Betrachtungen stellen folgende Kapitel jeweils spezielle Formen der Drahtantennen vor.

Zum Monobanddipol als einfachste Form der KW-Drahtantenne listet der Autor Richtwerte zur Gesamtlänge und der Dipolhälften auf. Aus einem Monobander wird eine Mehrband-Drahtantenne durch Parallelschaltung verschieden langer Dipolhälften. Interessant sind die gefalteten Dipole sowohl in Mono- als auch in Form einer Mehrbandantenne. Mechanisch verkürzte Dipole können mittels Verlängerungspulnen oder Endkapazitäten auf die benötigte elektrische Länge gebracht werden. Das Kapitel zeigt, wie es geht.

Mehrbanddipole mit Sperrkreise stellen komplexere Formen des Dipoles dar. Eine Tabelle nennt Daten zum Wickeln von Koaxialkabel-Sperrkreisen. Die W3DZZ wird als Beispiel dieser Antennenform eingeführt und um das 160-m-Band erweitert. Nichtresonante Dipole und Allband-Doppeldipole ermöglichen den Funkbetrieb auf allen Amateurfunkfrequenzen. Antennen von G5RV und ZS6BKW setzen hier Maßstäbe und sind sehr beliebt. Der Autor hat diese Antennen nachgebaut und berichtet über Resultate.

Das Buch wendet sich den asymmetrisch gespeisten Drahtantennen wie der Windom- und Stromsummenantenne zu und schließlich den endgespeisten Drahtantennen wie Fuchs- und Zeppelinantenne. Kleine Ganzwellenschleifen sind eine interessante Alternative auch für den Mehrbandbetrieb. Das Buch bietet Hinweise zum Aufbau und nennt Vor- und Nachteile. Wer weniger

Platz zur Verfügung hat, wird das Kapitel über vertikale Drahtantennen besonders aufmerksam lesen. Man erfährt von Groundplanes und Mehrbandgroundplane-Antennen und einem vertikalen Flächenstrahler für die Bänder von 10 bis 30 MHz.

Funkamateure ohne Möglichkeit, auf dem Dach, Garten oder Balkon eine Antenne zu montieren, werden auf den Portabelbetrieb ausweichen müssen. Leichte Antennen sind gefragt, die man schnell in die Luft bringen kann. Das Kapitel zeigt zahlreiche Varianten und nennt Tipps zum Aufbau. Tabellen, Abbildungen und Fotos helfen bei der Realisierung aller Antennenformen. Besonders praktisch ist der „Inverted-V“ aufgehängte Multi-Monobanddipol von DK9SQ. Es ist ein leichter Ein-Band-Dipol, der je nach zu arbeitenden Band auf die benötigte Länge ergänzt wird und nur einen Aufhängepunkt verlangt.

Indoorantennen im Sinne des Buches sind Zimmer- und Unterdachdrahtantennen. Hier zeigt der Autor anhand verschiedener Antennenformen, dass man das Hobby selbst unter schwierigen Bedingungen nicht an den Nagel hängen muss. Für die SWL's nennt ein Kapitel L- und T-Antennen, deren Länge an die örtlichen Gegebenheiten angepasst werden sowie eine breitbandige Empfangsantenne. Natürlich ist auch jede Sendeantenne als Empfangsantenne geeignet. Das abschließende Kapitel des KW-Drahtantennenbuches dreht sich um Anpassgeräte, dem Wahl des Koaxialkabels, Steckern und Zubehör. Hier ist auf Qualität zu achten. Im Anhang stehen Bemessungstabellen für Dipole, Ganzwellenschleifen, diverse Umrechnungstabellen und eine kleine Formelsammlung.

Fazit: Ein sehr lesenswertes Buch für den Selbstbau von KW-Drahtantennen. Es geht detailliert auf sämtliche Antennenformen ein, nennt Fakten und Daten sowie viele Tipps und Tricks zum wasserdichten Aufbau. Verlag und Redaktion wünschen dem Leser viel Freude mit diesem Buch, das aus der Hand eines erfahrenen Autors stammt. Sie finde es auf der virtuellen DVD in der Rubrik *Lesestoff/KWAntBuch*.

Michael Wöste, DL1DMW

Mehrere Lösungen online Lösungen online erstellen

Zur Berechnung und Planung von Linkstrecken auf den UHF- und SHF-Amateurfunkbändern erhöht eine vorherige Streckenplanung die Chance, einen Link zu etablieren. Für SHF-QSOs ist die Planung das Mittel für ein erfolgreiches QSO. Ein wichtiger Indikator ist das Höhenprofil bzw. der Geländeschnitt. Im Web gibt es dazu verschiedene Anwendungen, die wir hier vorstellen.

Früher wurden Geländeschnitte oft zur Linkplanung im 70-cm-Band für Packet Radio genutzt. Heute sind es Linkstrecken für das Hamnet bei 6 GHz. Auch Abseits der Linkplanung ist ein Geländeschnitt wertvolles Mittel zur Planung eines QSO auf den höheren Bändern. Je höher die Funkfrequenz, desto mehr Planung ist vonnöten. Das Höhenprofil hilft auch dabei, herauszufinden, weshalb einige Gegenden von eigenen Wohnort nicht oder nur sehr schlecht erreichbar sind - sei es auf dem 2m-, 70cm- oder den höheren Amateurfunkbändern.

Google Maps

Mit Google Maps ist eine ungefähre Abschätzung des Geländeschnitts möglich: Legen Sie auf Google Maps den Start- und Zielpunkt einer Route – also den Ort des Senders und Empfängers – fest. Klicken Sie danach auf das Fußgängersymbol, damit die Route möglichst ohne Umwege zum Zielpunkt verläuft. Unterhalb der Routenbeschreibung ist nun das Höhenprofil zu sehen. Da die auf der Karte gezeigte Strecke nicht exakt der Luftlinie entspricht, sind die Angaben des Höhenprofils recht ungenau (Bild 2).

Wie hoch ein Ort über NN auf einer Route oder generell auf der Karte liegt, erfahren Sie, wenn Sie mit der rechten Maustaste auf

den gesuchten Punkt klicken und im erscheinenden Kontextmenü die Option *Was ist hier?* wählen. Google zeigt Ihnen Infos wie z.B. den Orts- oder Flurnamen sowie den Breiten- und Längengrad. Klicken Sie auf die Koordinaten und kopieren Sie die Angaben mit Strg+C in die Zwischenablage. Gehen Sie nun zu mapcoordinates.net (Bild 3) und kopieren in das dortige Suchfeld in der Landkarte die Koordinaten mit Strg+V ein und betätigen anschließend die Enter-Taste. Auf der Karte sehen Sie nun, wo sich die gesuchte Position befindet und erfahren zudem die Höhe über NN.

Bei gedrückter Maustaste lässt sich der gesetzte Punkt verschieben, um Höhen von Orten in der Nähe zu betrachten.

Mobile Radio

Mobile Radio [3] von VE2DBE ist sehr bekannt und leistet weit mehr, als nur das Höhenprofil zu berechnen. Es wendet sich an Funkamateure und andere Funkinteressierte. Man kann Mobile Radio als Windows-Programm auf den PC laden und installieren – oder alternativ die Online-Version benutzen. Letzteres spart Platz auf der Festplatte und die

Mühen der Installation. Die Online-Version führt die Anwendung auf dem Webserver aus und präsentiert Ergebnisse im lokalen Browser. Diese Version werden wir benutzen.

Man muss sich bei [3] anmelden und nach dem Einloggen im Hauptmenü mit *New Site* zwei neue Standorte für Start- und Zielpunkt definieren. Anhand dieser Punkte kreiert man nun einen neuen Link (*New Link*). Ist das geschehen, wählt man ihn an und dort im dortigen Menü klickt man auf *View this link* (Bild 4). In Mobile Radio definiert man zudem die Sende- und Empfangsantenne und bestimmt anhand der Sendeleistung und Topografie die Ausbreitung rund um die Antenne, als *coverage* bezeichnet (siehe Bild 5).

Geländeschnitt mit ArcGIS:

Laden Sie die Karte über die Webadresse [4]. Sie sehen die USA-Karte, genauer den Iowa Trail. Zoomen Sie nun mit dem Mausrad heraus und steuern Europa und ihre Wohngegend an. Die Punkte auf der Karte, die zur Berechnung des Höhenprofils dienen, setzt man so: Klicken Sie in dem grauen Balken („Iowa Trail“) auf das Stiftsymbol, klicken mit der Maus zuerst den Start- und dann den Zielpunkt an. Mit einem Doppelklick beenden Sie

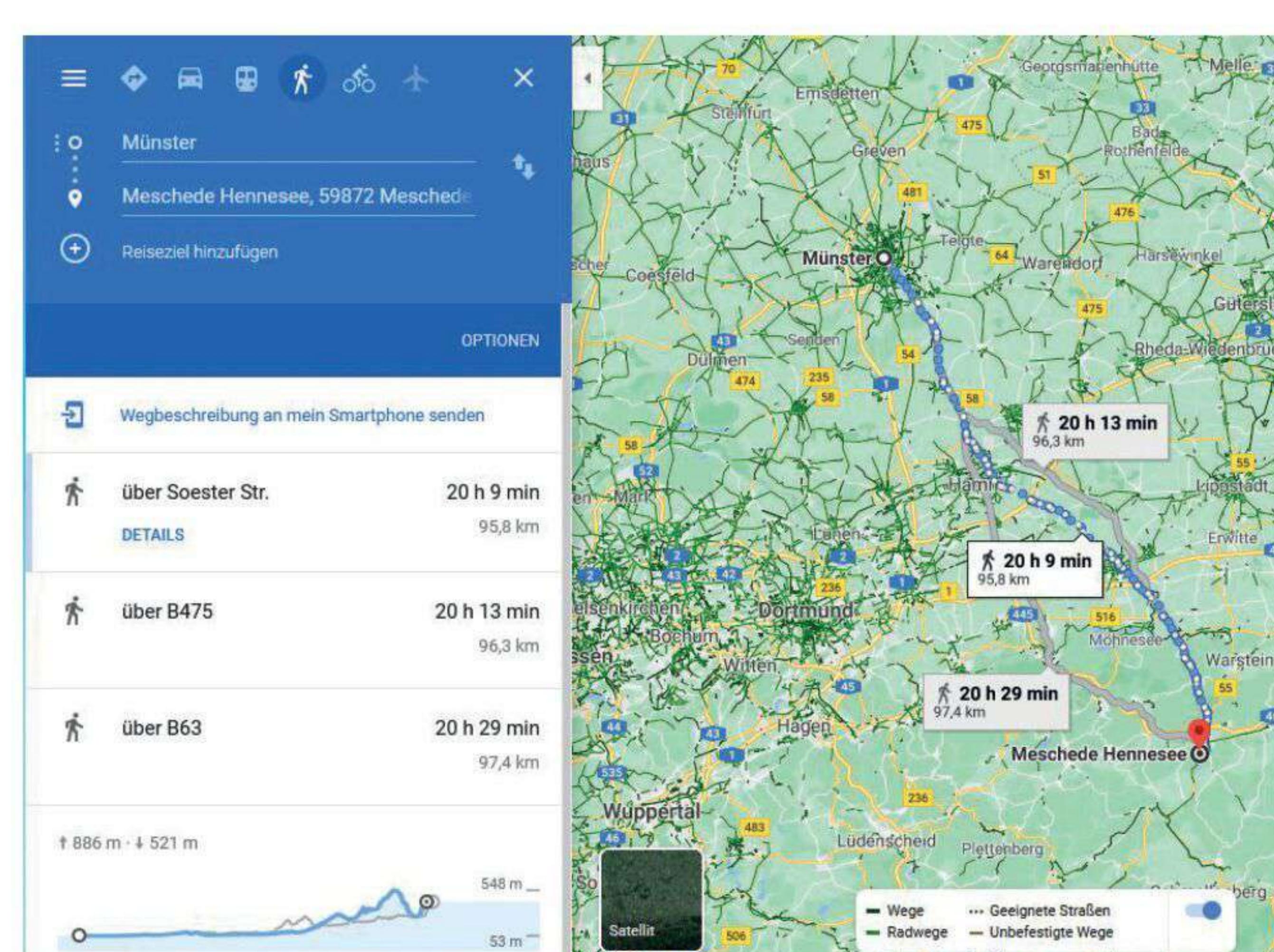


Bild 2: Google Maps ermittelt das Höhenprofil einer Route. Gegenüber einer Luftlinie zwischen Sender- zum Empfangspunkt ergibt sich hier lediglich eine Annäherung.

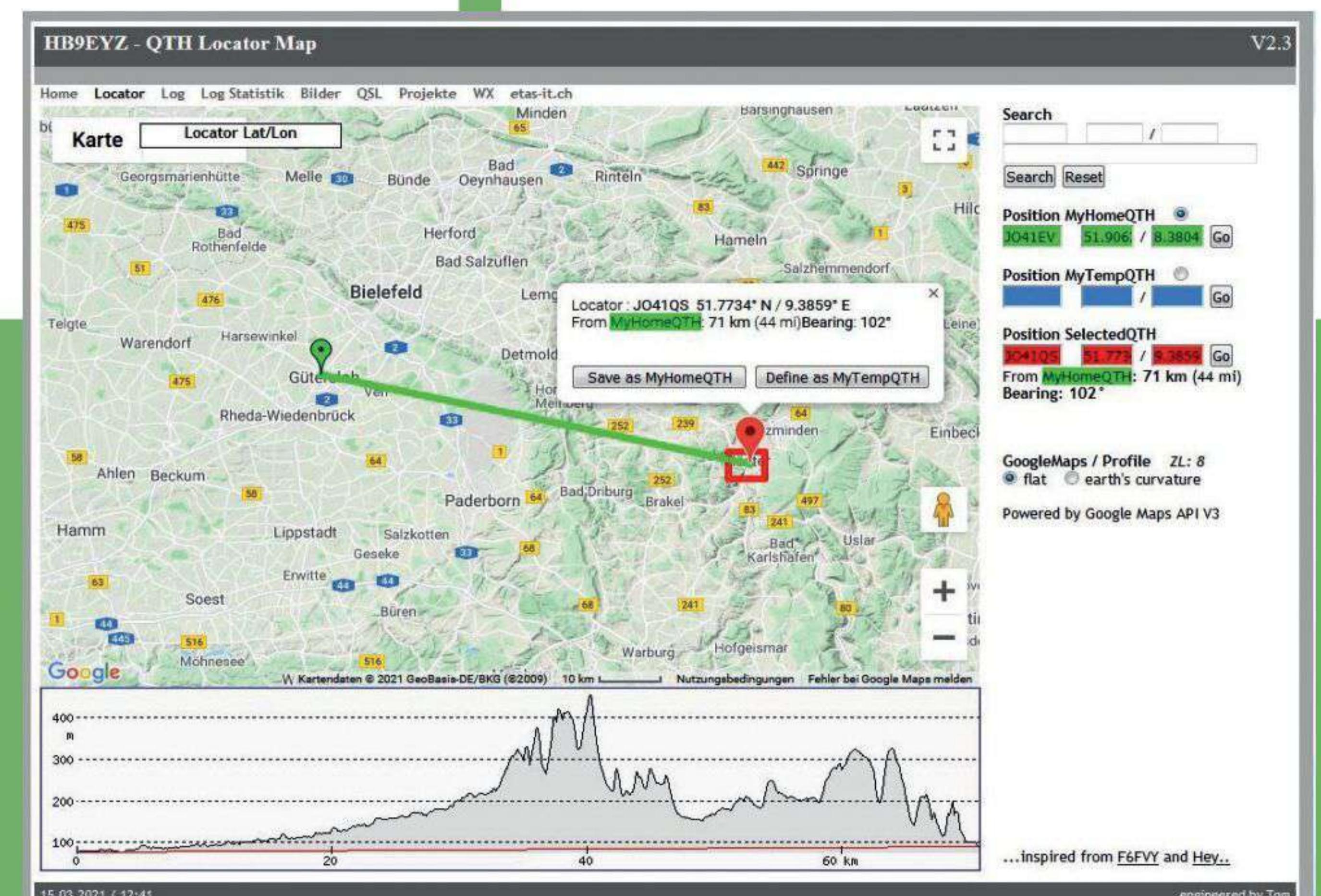


Bild 1: Locator-Map und Höhenprofil auf der Webseite von HB9EYZ.

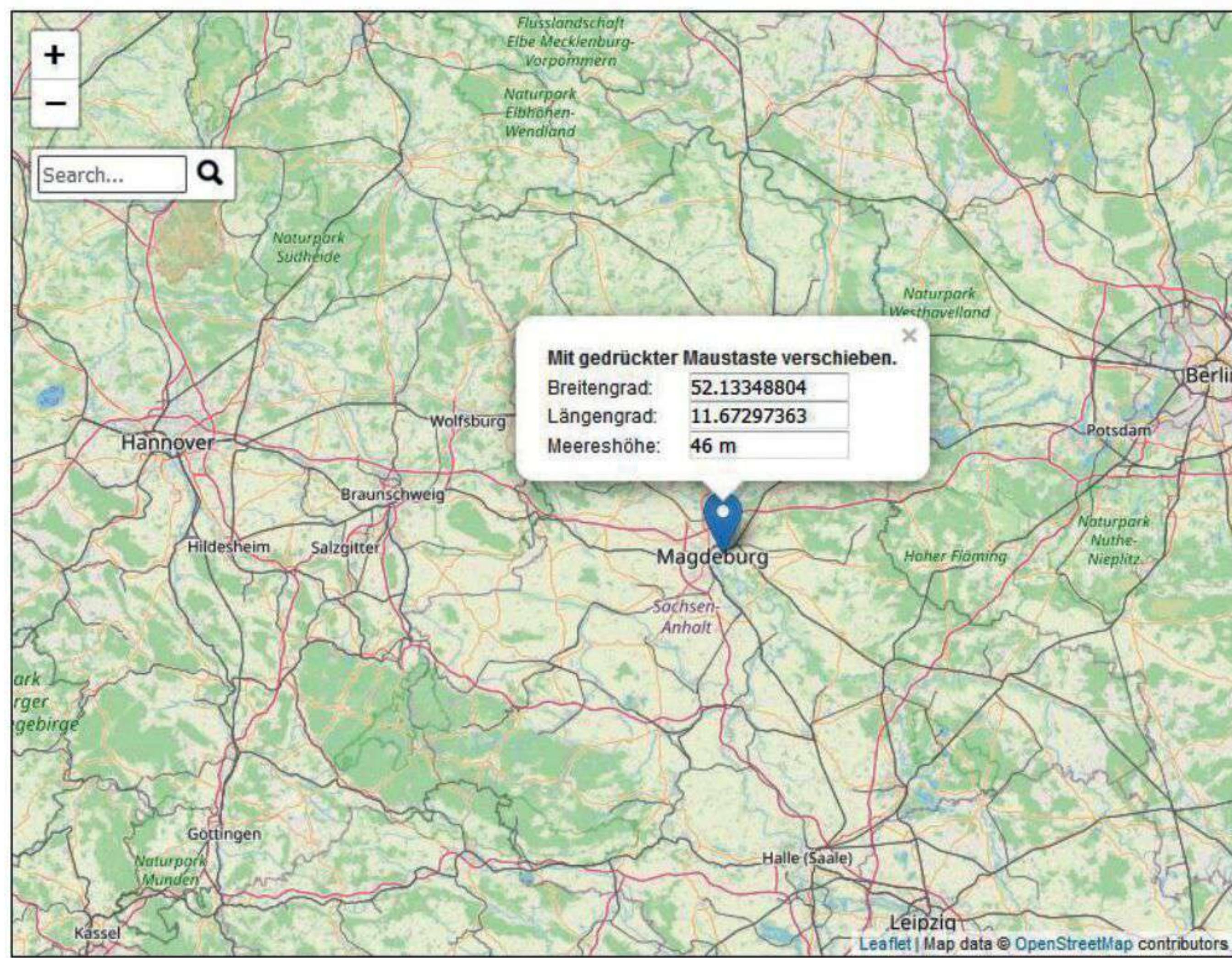


Bild 3: Mapcoordinates.net: Kopiert man eine Koordinate in das Suchfeld, informiert die Landkarte über die Höhe über NN.

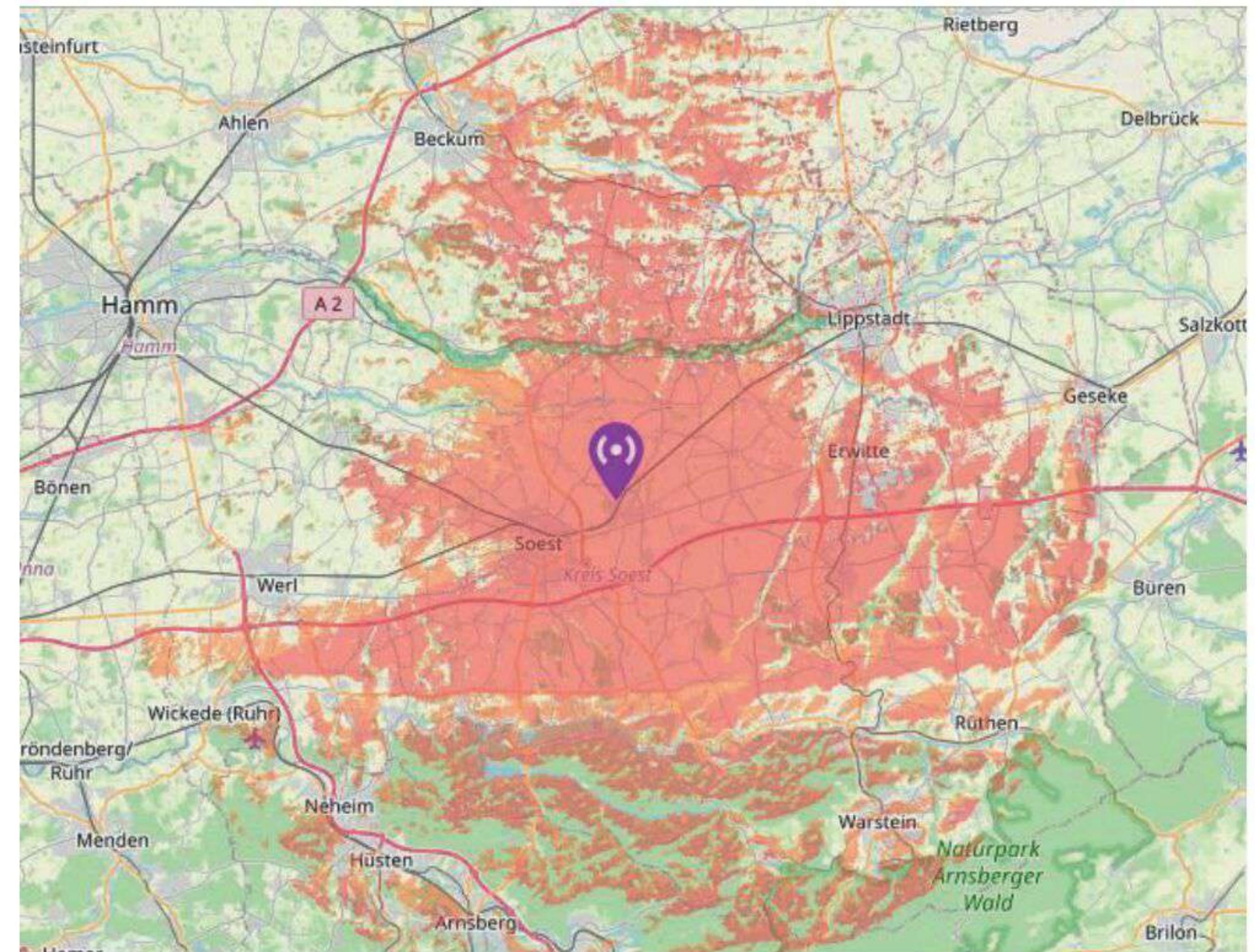


Bild 5: Mobile Radio beantwortet die Frage: Welche Flächenabdeckung hat mein HF-Signal?

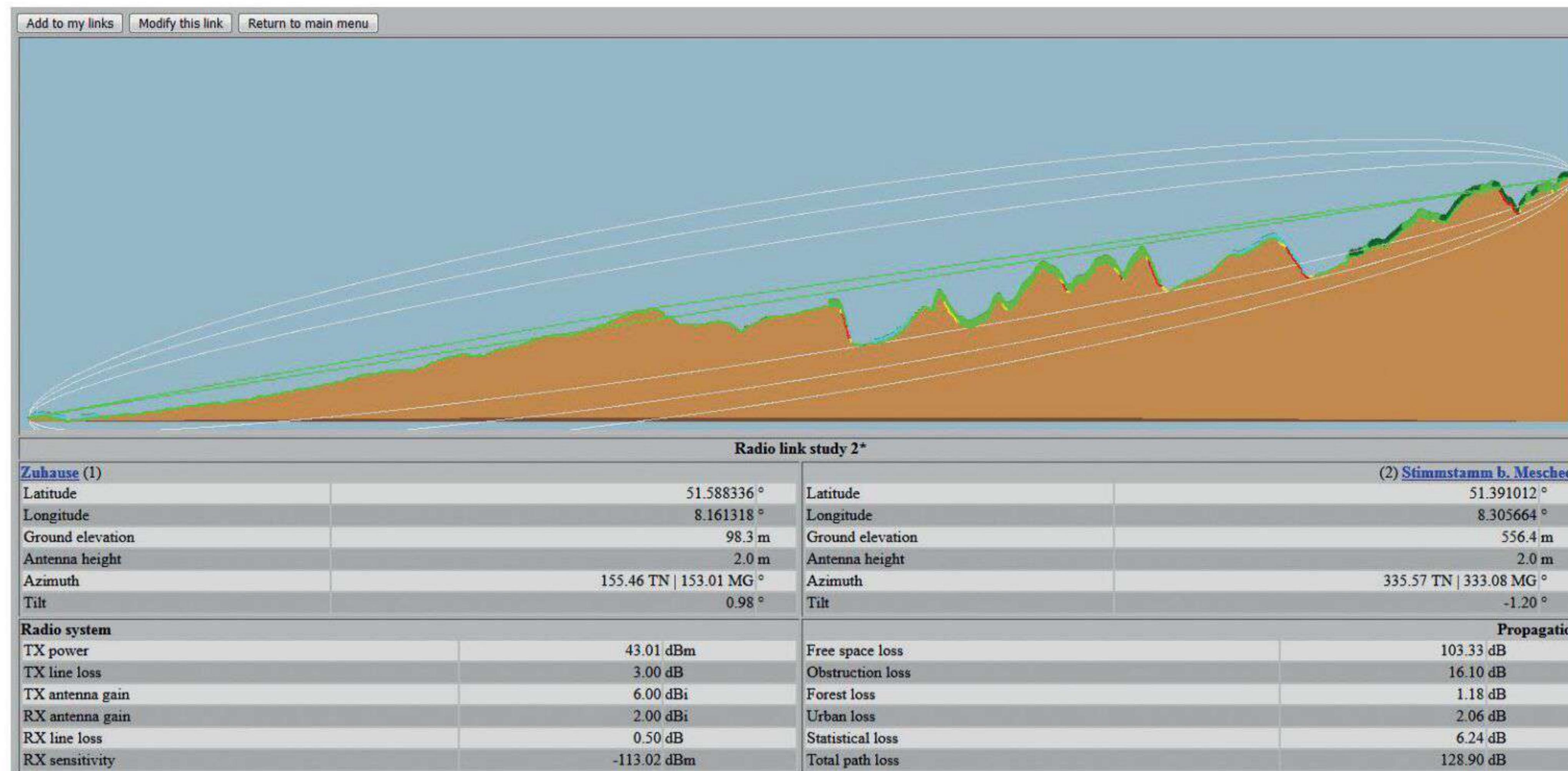


Bild 4: Mobile Radio ermittelt neben dem Höhenprofil die Fresnelzone und berechnet die Signalverluste auf dem Weg zum Ziel (total path loss)

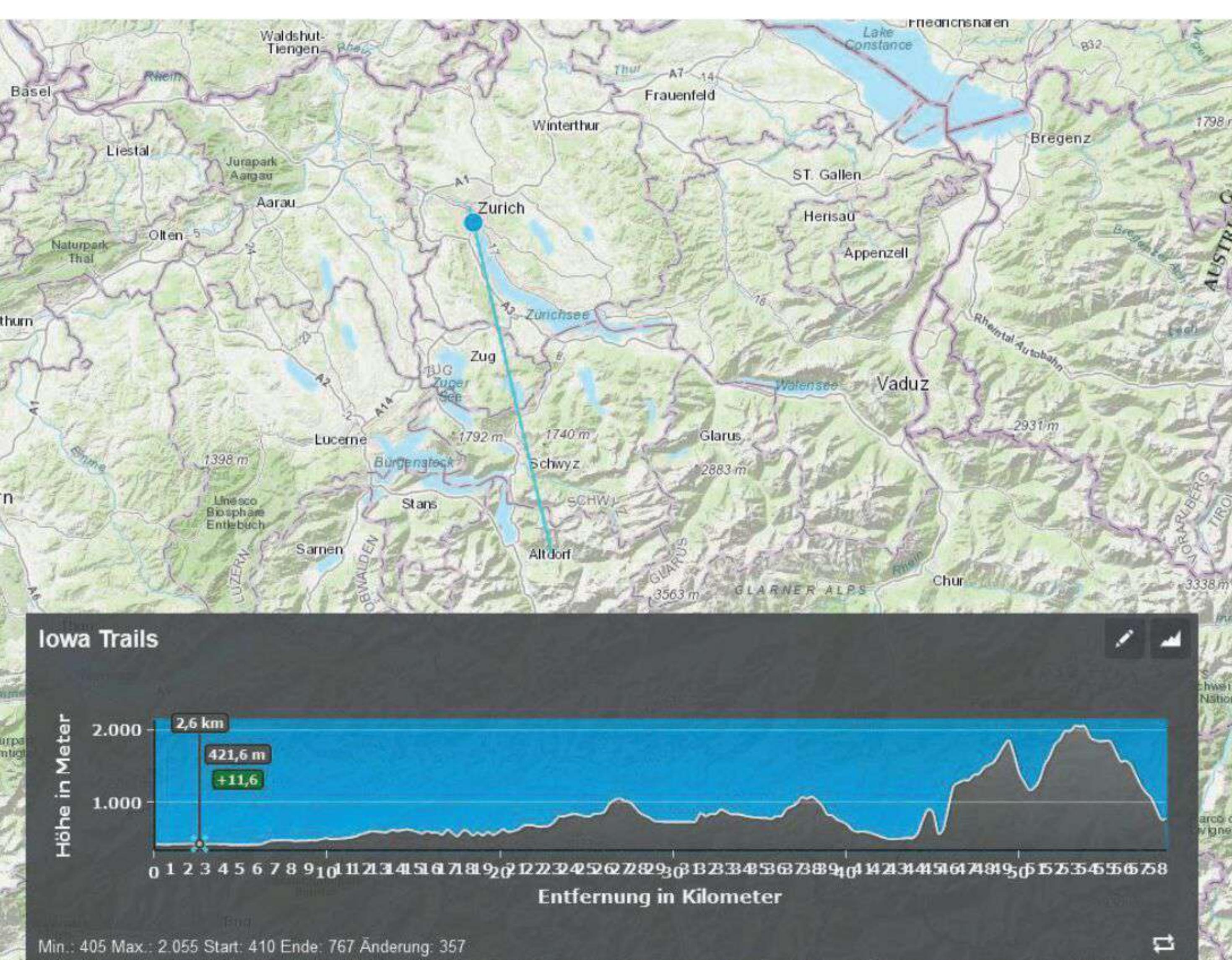


Bild 6: Höhenprofil mit ArcGIS.

das Höhenprofil (siehe Aufmacherbild). Für beide Positionen werden der QTH-Locator berechnet und die Beamrichtung angezeigt. Ein weiterer Mausklick in die Landkarte verschiebt die zweite Position und bewirkt eine Neuberechnung des Geländeschnitts. Ein Klick auf das Höhenprofil zentriert die Landkarte auf eben diese Position.

Google Earth

Das Programm – hier für Windows – lässt sich zur Link- und QSO-Planung einsetzen. Mit wenigen Klicks lässt sich in Google Earth ein Pfad anlegen, der dann auch auf Wunsch ein Höhenprofil enthält. Dieses Höhenprofil lässt sich speichern, um beim nächsten Programmstart wieder aufgerufen zu werden. Starten Sie Google Earth Pro und navigieren auf dem Globus zur gewünschten Gegend. Zoomen Sie weit genug in das Bild hinein, damit Sender- als auch Empfängerstandort auf die Karte passen. Nun öffnen Sie das Menü *Tools* und wählen das *Lineal* aus. Es öffnet sich das Lineal-Fenster. Wählen Sie den Karteireiter *Pfad* und stellen unter *Länge* die Maßeinheit auf Kilometer. Der Mauszeiger hat sich in ein Fadenkreuz verwandelt. Klicken Sie damit zuerst auf den Sender, dann auf den Ort des Empfängers. Es muss sich dabei nicht um einen Ortschaft oder Straße handeln, Sie können die Punkte überall auf die Karte platzieren.

die Eingabe. Sofort berechnet das Programm das Höhenprofil. Fahren Sie mit der Maus über die Grafik, zeigt eine vertikale Markierung für diesen Punkt die Höhe über NN, die Höhendifferenz zum Startpunkt (Sender) und die Entfernung zum Start in Kilometern (siehe Bild 4).

HB9EYZ: Locator Map und Geländeprofil

Eine schnelle und einfache Alternative zur Erstellung eines Höhenprofils bietet die Webseite von HB9EYZ [5]. Punkte in der Karte setzt man mit einem Doppelklick. Die erste Position kann man als Heimat-QTH übernehmen. Nach Setzen der zweiten Position erscheint sofort

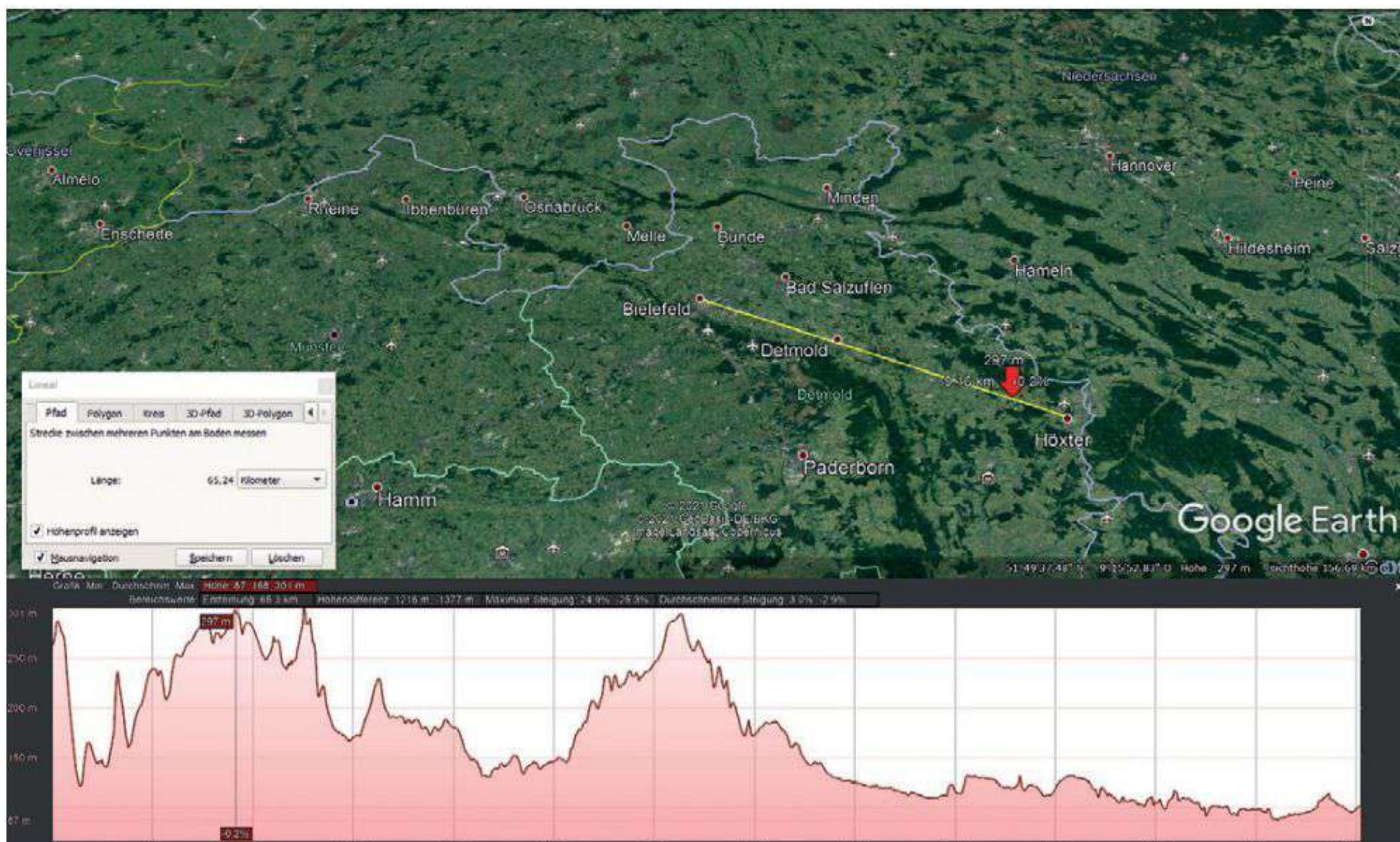


Bild 7: Auch mit Google Earth Pro kommt man dank der Lineal-Funktion schnell zu einem Höhenprofil.

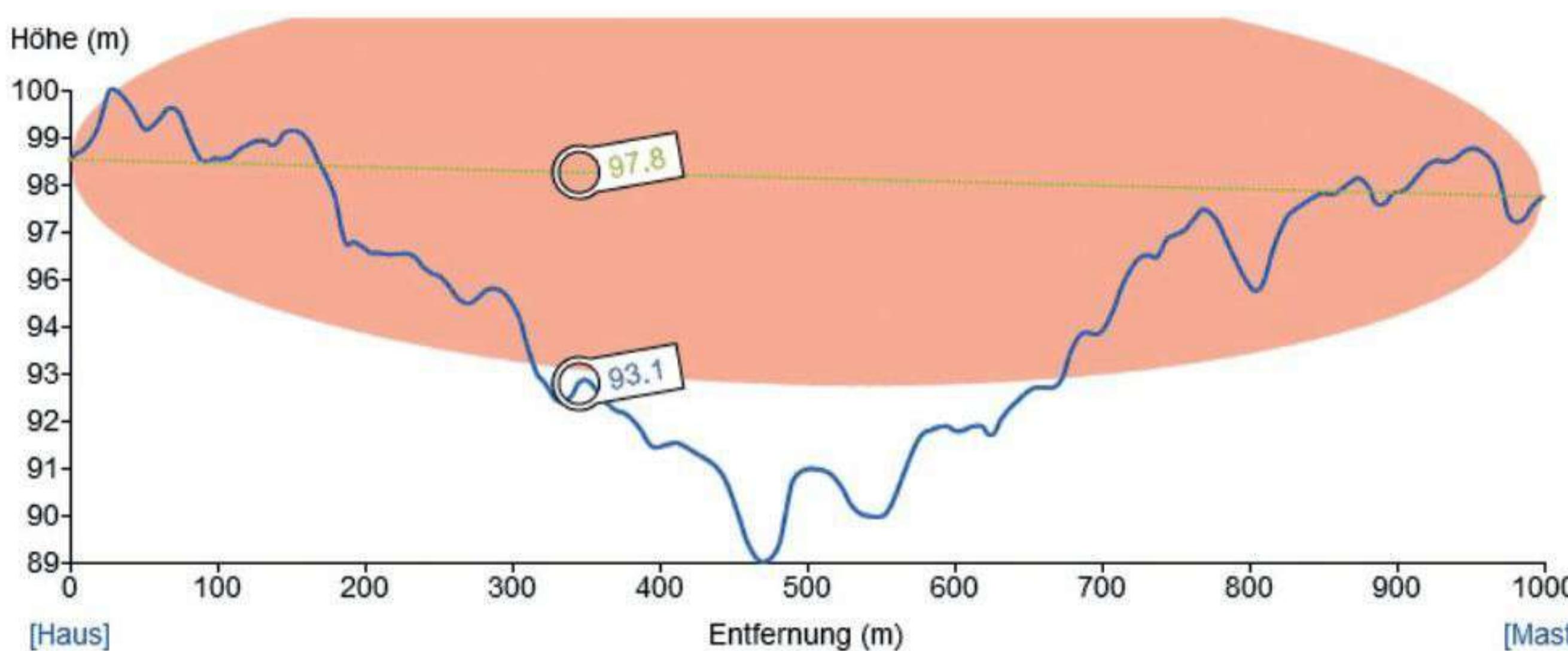


Bild 8: Höhenprofil zwischen eigenem Standort und dem nächsten LTE-Sendemasten. Im Expertenmodus bestimmt man unter anderem die Frequenz.

Nun sieht man das Höhenprofil des Pfades unterhalb der Karte in einem neuen Bildschirmbereich (siehe Bild 7). Verändern Sie die Position des Senders oder Empfängers im Kartenausschnitt, aktualisiert sich das Höhenprofil automatisch.

Auf Wunsch klickt man in dem Fenster *Lineal* auf *Speichern*, gibt dem Höhenprofil eine Bezeichnung und speichert es ab. Anschließend sind gespeicherte Höhenprofile unter *Temporäre Orte* zu sehen. Um sie dauerhaft zur weiteren Verwendung zu speichern, sollten Sie sie in *Meine Orte* verschieben.

Mastes (!) mit der Maus neu positionieren. Unterhalb der Karte ist das Höhenprofil einzusehen, dass sich nach jeder Neuposition des Standorts bzw. des Mastes aktualisiert. Im „Expertenmodus“ sind weitere Parameter wie Höhe des Mastes, Höhe der eigenen Antenne und eine der LTE-Frequenzen einstellbar. Bild 8 zeigt ein Höhenprofil mit berechneter Fresnelzone.

Literatur/Verweise:

- [1] Google- Maps: <https://www.google.de/maps/>
- [2] Mapcoordinates: <https://www.mapcoordinates.net/de>
- [3] Radio Mobile Online: https://www.ve2dbe.com/rmonline_s.asp
- [4] ArcGIS: <https://www.arcgis.com/apps/Profile/index.html>
- [5] Locator Map und Geländeprofil: https://www.hb9eyz.ch/g_locator.php
- [6] Google-Earth download: <https://www.google.de/earth/download/gep/agree.html>
- [7] LTE-Höhenprofil: <https://www.lte-anbieter.info/gelaende/hoehenprofil.php>

Michael Wöste, DL1DMW

LTE-Höhenprofil

Abseits der Amateurfunkfrequenzen nutzt man gern Höhenprofile, um z.B. den Zugriff zu dem nächsten LTE-Masten zu testen. Ein Höhenprofil zwischen Standort/Wohnort und dem nächsten LTE-Sendemasten bietet [7]. Nach Eingabe von Straße und Wohnort geht es los. Auf der Karte, die man als Resultat betrachtet, kann man sowohl den eigenen Standort als auch den des

TIPP: Flüssiger schreiben mit Android

Tablet-PC und Smartphone mit Android-Betriebssystem verfügen über einen USB-Anschluss. Mit einem USB-OTG-Kabel gelingt die Kommunikation mit verschiedenen USB-Geräten. Das funktioniert auch mit einer Tastatur. Dann wird das Schreiben längerer Texte zur Freude.

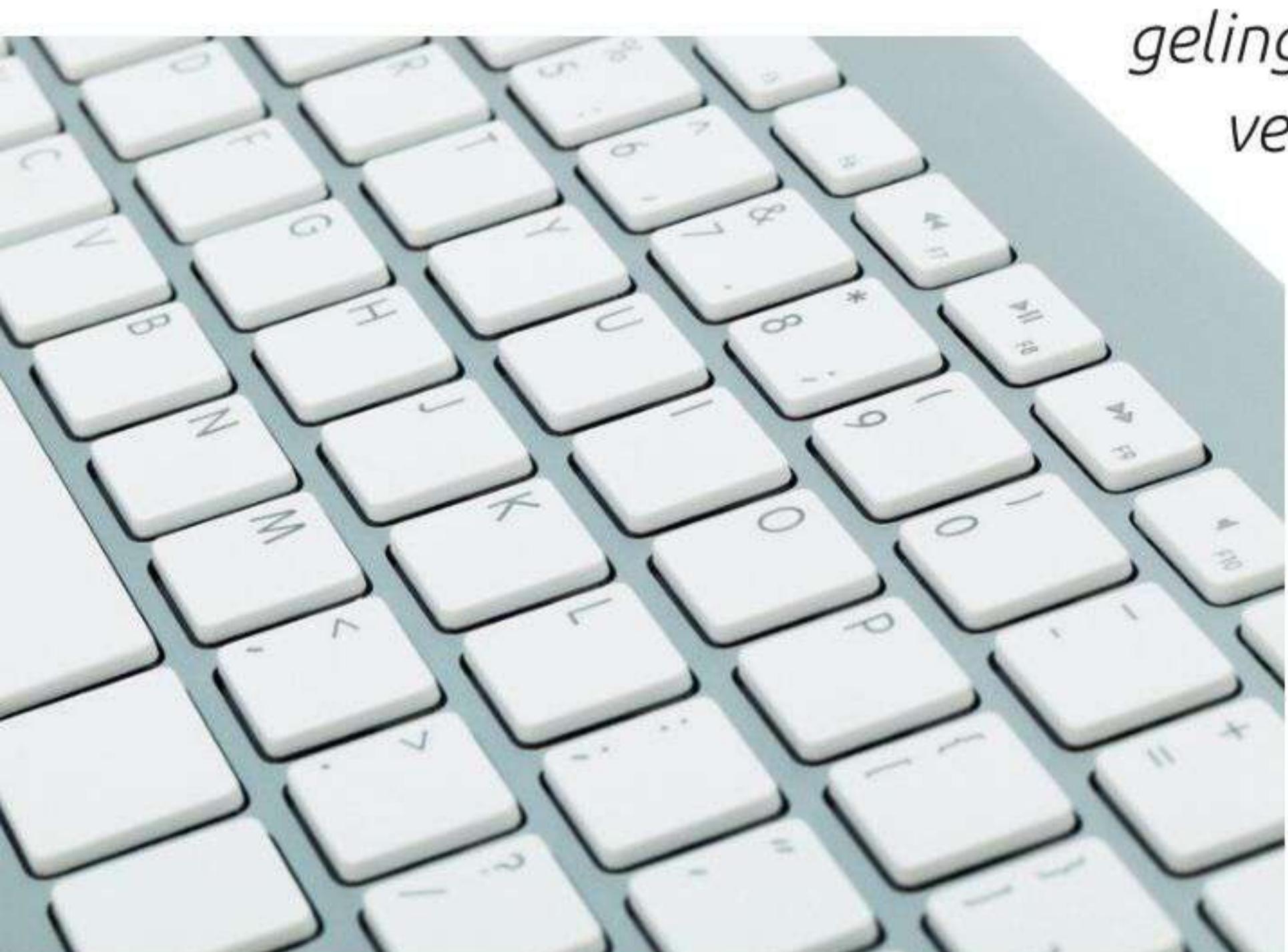


Bild 1: Mini-USB-Tastatur, praktisch für Vielschreiber mit Android Tablet-PC und Smartphone.

Eine „ausgewachsene“ Tastatur belegt viel Fläche und ist sperrig beim Transport. Es gibt auch kompakte Varianten, die nur wenig größer als ein Smartphone und kleiner als ein Tablet-PC sind. Mit dieser Kombination aus Gerät und Tastatur ist das Schreiben längerer Texte abseits aller Steckdosen möglich – nicht nur die Mail an die Oma, auch das Programmieren beispielsweise für den Arduino (mit der ArduinoDroid App) ist dann keine Kunst mehr.

Mit dicken Fingern bedient sich eine externe Tastatur wesentlich besser als eine fummelige Bildschirmtastatur auf dem Bildschirm eines Smartphones, auf der man gern zwei Tasten zugleich erwischte. Noch ein Vorteil: Es steht mehr Platz zum Edieren des Textes auf dem Bildschirm zur Verfügung, da die Displayfläche der eingeblendeten Tastatur entfällt. In den Android-Einstellungen kann

man die Tastenbelegung wählen.

Über einen USB-Hub lassen sich zwei oder gar mehrere USB-Geräte an ein Tablet bzw. Smartphone anschließen. Das kann z.B. die Tastatur und ein Arduino zur Programmierung desselben unterwegs oder eine Tastatur und ein USB-Stick sein. Damit der Akku des USB-Host nicht zu stark belastet wird – er müsste alle Geräte versorgen – und damit die Akkulaufzeit des Gerätes nicht allzu stark sinkt, wählt man einen aktiven USB-Hub. Letzterer besitzt eine eigene Spannungsversorgung über ein Steckernetzteil, abseits aller Steckdosen am Strand und auf dem Berg weicht man auf eine Powerbank aus. Nachteil: Etwas Kabelsalat rund um das Smartphone oder das Tablet-PC muss man zu Gunsten gestiegener Schreibqualität akzeptieren.

Michael Wöste, DL1DMW

Für den Datentransfer zwischen Android Tablet und Desktop-PC nutzen viele Anwender als Speichermedium, als „Zwischenwirt“, einen USB-Speicherstick, den man über ein USB-OTG-Kabel anschließt. Vom Tablet-PC kopiert man die Dateien zuerst auf den USB-Stick,

steckt ihn an dem PC um und kopiert die Daten auf den PC. Das Tablet fungiert dabei als USB-Host. Aufgemerkt: An einen USB-Host kann man eine USB-Kabeltastatur anschließen. Das Android-Betriebssystem besitzt die notwendigen Fähigkeiten.

Links to Old Radio Frequency Books and Magazines - Archive Collection

<http://www.qsl.net/va3iul>

Note: Some pdf files are very large. Always use "Save Link As..." option.

- [Radiotron Designer's Handbook \(1941\) 3rd Edition - F. Langford-Smith](#)
- [Radiotron Designer's Handbook \(1953\) 4th Edition - F. Langford-Smith](#)
- [Radio Engineers' Handbook \(1943\) - F. E. Terman](#)
- [Microwave Receivers \(1948\) - Van Voorhis](#)
- [Microwave Transmission Line Impedance Data \(1972\) - Gunston](#)
- [Microwave Transmission Circuits \(1948\) - Ragan](#)
- [Advances in Microwaves \(1966\) - Young - Volume 1 - Volume 2](#)
- [Technique of Microwave Measurements \(1947\) - Montgomery](#)
- [Transistor Superhet Receivers \(1960\) - Sir Clive Sinclair](#)
- [Radio Receivers \(1970\) - Barkan, Zhadanov](#)
- [Antennas - Theory and Practice \(1952\) - Schelkunoff, Friis](#)
- [Advanced Antenna Theory \(1952\) - Schelkunoff](#)
- [Radio Telephony for Amateurs \(1922\) - Ballantine](#)
- [73 Vertical, Beam and Triangle Antennas \(1970\) - E. Noll W3FQJ](#)
- [73 Dipole and Long-Wire Antennas \(1969\) - E. Noll W3FQJ](#)
- [A Guide to Amateur Radio - 12th Edition \(1966\) - J. P. Hawker G3VA](#)
- [The Gunnplexer Cookbook \(1981\) - R. Richardson W4UCH](#)
- [Basic Radio - The Essentials of Electron Tubes and Their Circuits \(1942\) - J. B. Hoag](#)
- [Electrical Communication \(1950\) - A. Albert](#)
- [Electronics for Engineers - 1st ed. \(1945\) - J. Markus, V. Zeluff](#)
- [Electronics for Communications Engineers \(1952\) - J. Markus, V. Zeluff](#)
- [Electronics for Engineers \(1945\) - J. Markus, V. Zeluff](#)

Bild 1: Die Webseite listet gut 200 Links zu Büchern mit Bezug zu HF-Technik, Radio und Amateurfunk auf.

amateur
FUNK

ELIMINATING MAN-MADE INTERFERENCE

by JACK DARR

Here are the solutions to all your noise and interference problems in broadcast, communications, and TV receivers.

A. J. H. Darr PHOTOFAC PUBLICATION

Bild 2: Beweis: Schon in den 60er Jahren kämpfte man gegen Störungen elektrischer Geräte.

Old Radio Books: Lesevergnügen aus der Vergangenheit

Auf [1] hat YO3DAC/VA3IUL geschätzt zweihundert Links zu alten Büchern aus den Jahren 1908 bis zur Jahrtausendwende zusammengestellt. Selbst wenn einige Links ins Leere gehen, lohnt ein Besuch dieser Linkliste englischer Amateurfunkliteratur.

Das Funkamateure schon 1960 dieselben Probleme lösen mussten, zeigt das Buch „Eliminating Man Made Noise“ von Jack Darr. Dieses Werk – und viele weitere – findet man auf der Webseite als lange Linkliste. Die Bücher stehen allesamt als PDF-Dateien zum Download bereit. Ältere, auf Papier vorliegende Werke wurden in dieses weltweit gängige Dokumentenformat überführt.

Das Themenspektrum der Linkliste ist weit gespannt: Anfängerliteratur zum Radiobau, Tipps

zur Benutzung von Messgeräten, Schaltbilder, die vergangene Baulementen nutzen, und Literatur zur Wellenausbreitung: Hier wird man sicher fündig.

In Gary L. Frosts Buch „Early FM“ ist beispielsweise zu lesen: „Im Jahr 1933 erteilte das US-Patentamt Armstrong Patente für sein System des „Breitband“-Frequenzmodulationsradios. Mehr als ein Jahrzehnt zuvor hatte jeder andere FM als unpraktisch aufgegeben, aber Armstrongs System verblüffte die Welt, indem es statische Störungen

unterdrückte und den Klang mit weitaus größerer Genauigkeit wiedergab als AM-Radio...“ ([3]). Interessant ist zu lesen, dass es zu dieser Zeit erste Versuche mit FM gab. Die in den USA arbeitenden Mittelwellensender arbeiten heute noch traditionell mit AM. Wer sich zur Amplitudenmodulation informieren möchte, liest das Buch „Amplitude Modulation“ von Alexander Schure. Dieses Buch kostete damals \$ 1,25. Freunde alter Radios finden Anleitungen zur Reparatur derselben: „Radio Receiver Servicing 2nd“ ist ein Beispiel oder „Servicing Transistor Radios“ aus dem Jahre 1959. Sicherlich war es eine wertvolle Hilfe für die Reparatur der gerade aufkommenden portablen Radios, die man auch zum Picknick nehmen konnte.

Auch Biografien findet man, zum Beispiel die von James Clerk Maxwell unter dem Titel „The Man who changed everything“ aus dem Jahr 2003. Die Webseite listet zu

Marconi gleich mehrere Bücher auf und auch Tesla darf nicht fehlen.

Freunde alter Radioröhren finden eine reiche Auswahl an Schaltungsvorschlägen, beispielsweise in dem Buch „Radio Circuits“ von Miller aus dem Jahr 1944 oder „More simple Radio Circuits“, das A. T. Collins 1965, Chefredakteur des englischen „Practical Wireless Magazine“, verfasst hat. Wer sich eher für die Funktionsweise der Röhren interessiert, dem dürfte das Buch „Inside Radio Tubes“ von John F. Rider aus dem Jahr 1945 gefallen.

Habe ich Ihnen Lust auf alte Radiobücher gemacht? Dann nichts wie hin zu [1]. Auf der Webseite des OM [2] findet man nicht nur die Literatur-Links, sondern eine riesige Menge aktueller Information und Schaltungen zur HF-Elektronik, interessante Bauprojekte und weitere Linklisten für Afu-Software, CAD-Programme und vieles mehr.

Literatur/Verweise:

[1] Englischsprachige Afu-Bücher: https://www.qsl.net/va3iul/Files/Old_Radio_Frequency_Books.htm

[2] Homepage von VA3IUL: <https://www.qsl.net/va3iul/>

[3] Aus dem englischen Übersetzt mit: <https://www.deepl.com/translator>

Michael Wöste, DL1DMW

Speziell für batteriege-triebene Mikroelektro-nik wurde dieser Timer geschaffen. Zyklisch schaltet er die Span-nungsversorgung einer Last ein und aus. Wir wagen einen Blick auf diesen interessanten IC.

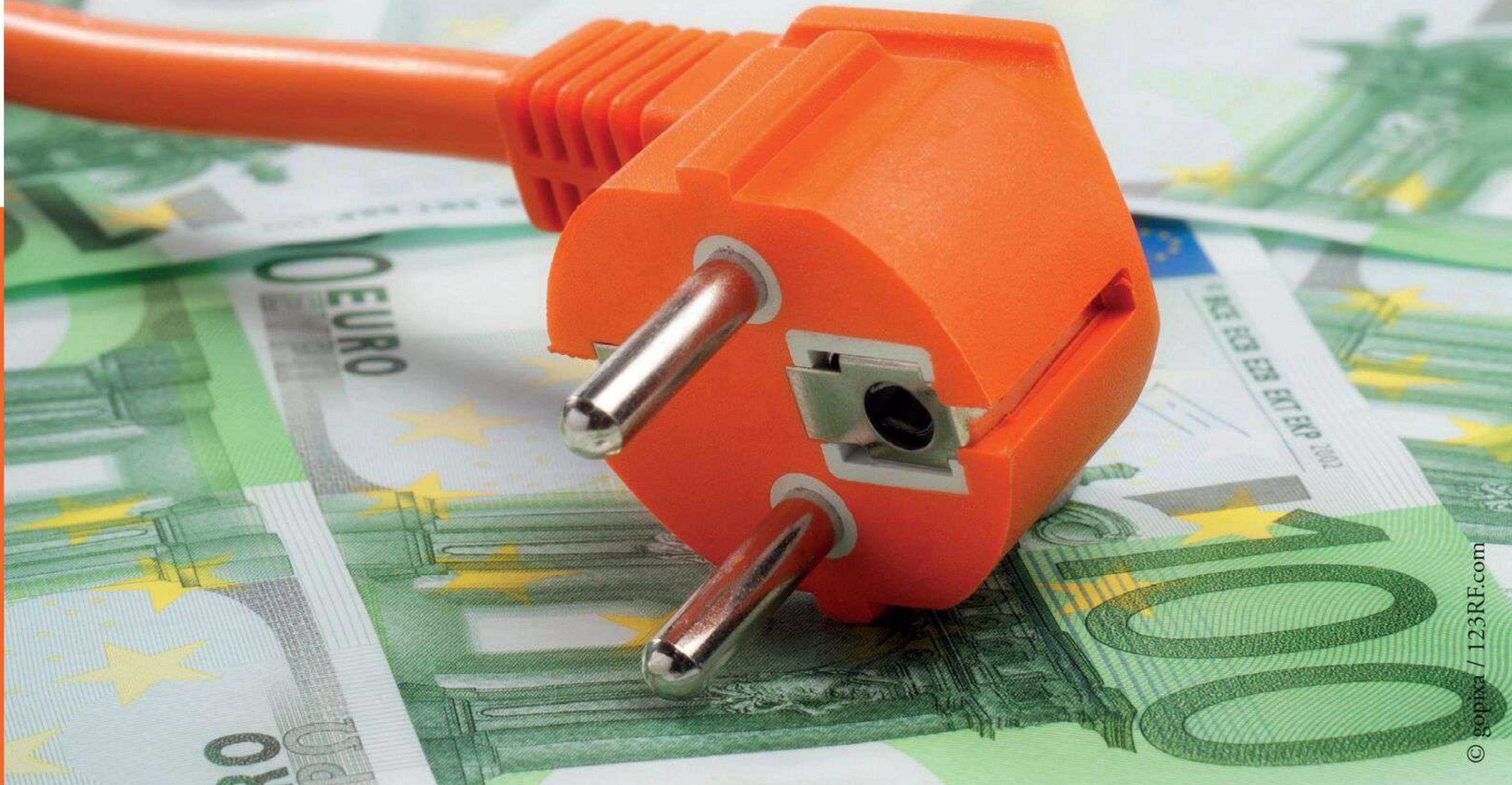
Eine klassische Internet-der-Dinge-Anwendung ist die von einem Mikrocontroller gesteuerte Schaltung, die hin und wieder – in zyklischen Abständen, sagen wir von zehn Minuten – eine bestimmte Aufgabe ausführt. Das kann typisch die Abfrage eines Sensors sein, dessen Messwert auf eine beliebige Weise, oft per Funk, übermittelt wird. Es könnte sich vielleicht um eine Temperatur, einen Druck oder einen beliebigen anderen Messwert handeln. Die Schaltung läuft in der Regel stundenlang durch und lediglich alle 10 Minuten hat der Mikrocontroller für einige Sekunden etwas zu tun. Danach ist er für mehr als 9 Minuten ohne Funktion, benötigt aber Strom. Wird das Gerät von einer Batterie gespeist und diese weist genug Kapazität auf, ist dagegen nichts einzuwenden. Doch: Wie man es wendet oder dreht: Es ist Stromverschwendungen.

Nun kommt der Timer TPL5110 ins Spiel: Er sorgt dafür, dass die angesprochene Schaltung nur dann mit Spannung versorgt wird,

Bild 1: Pinbelegung des TPL5110 Nano Power Timers. Quelle: Datenblatt.

PIN		TYPE ⁽¹⁾	DESCRIPTION	APPLICATION INFORMATION
NO.	NAME			
1	VDD	P	Supply voltage	
2	GND	G	Ground	
3	DELAY/ M_DRV	I	Time interval set and manual MOSFET Power ON	Resistance between this pin and GND is used to select the time interval. The manual MOSFET power ON switch is also connected to this pin.
4	DONE	I	Logic Input for watchdog functionality	Digital signal driven by the µC to indicate successful processing.
5	DRV	O	Power Gating output signal generated every tip	The Gate of the MOSFET is connected to this pin. When DRV = LOW, the MOSFET is ON.
6	EN/ ONE_SHOT	I	Selector of mode of operation	When EN/ONE_SHOT = HIGH, the TPL5110 works as a TIMER. When EN/ONE_SHOT = LOW, the TPL5110 turns on the MOSFET one time for the programmed time interval. The next power on of the MOSFET is enabled by the manual power ON.

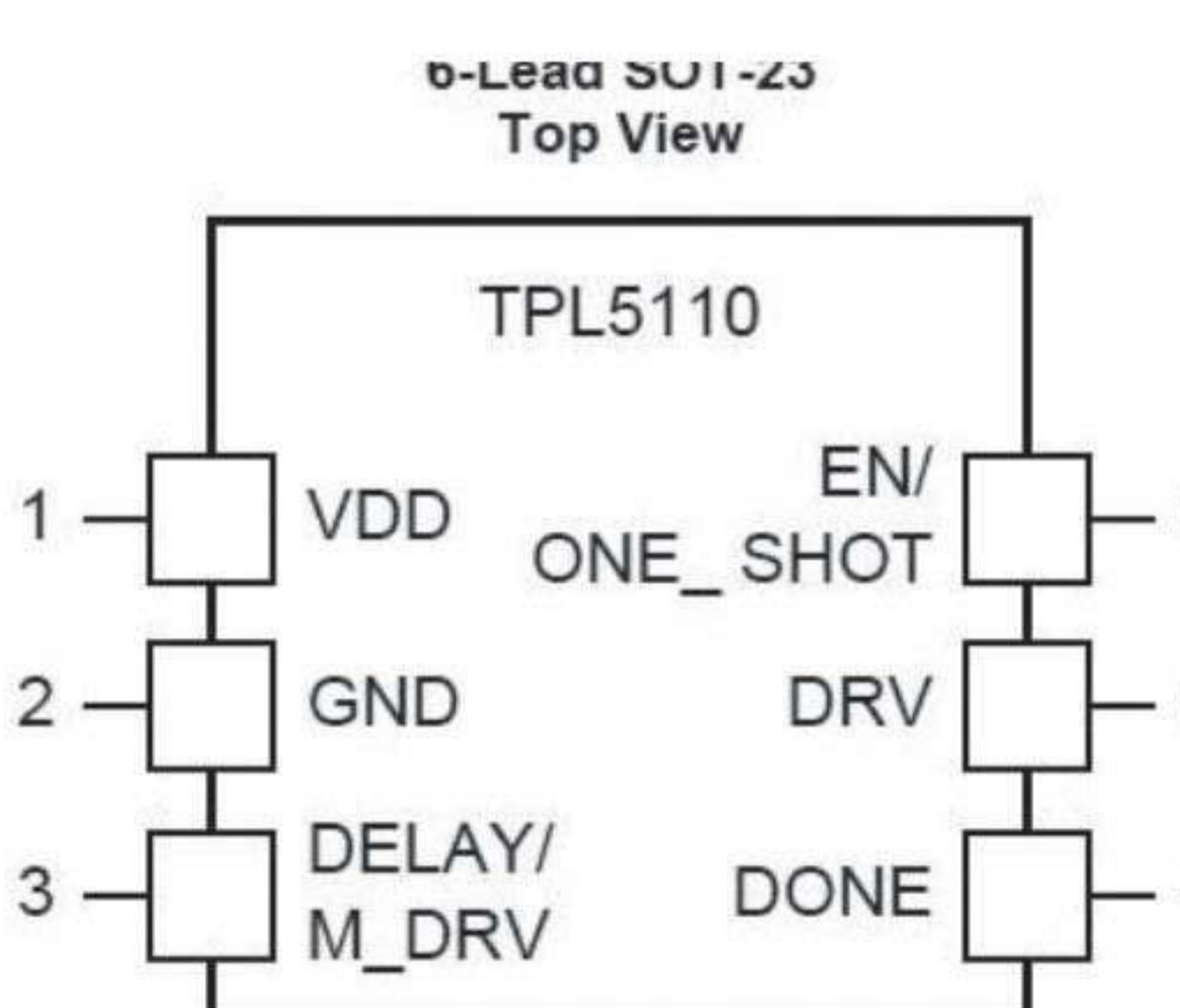
(1) G= Ground, P= Power, O= Output, I= Input.



Der Stromspar-Timer: Nano Timer TPL5110

wenn sie wirklich benötigt wird – im Beispiel alle zehn Minuten. In der Zwischenzeit ist die Schaltung aus, es läuft allein der Timer und der benötigt nur sehr wenig Strom: Laut Datenblatt begnügt sich der TPL5110 bei 2,5 Volt Betriebsspannung mit nur 0,000.000.035 Ampera, das sind winzige 35 nA. Meine vorhandenen Messmittel reichen

nicht aus, derart geringe Ströme sicher zu messen. Im Datenblatt entnimmt man einem Diagramm, dass der vom Timer benötigte Strom bei 3,3 Volt bei ca. 38 nA liegt und bei 5 Volt beträgt er ca. 44 nA (Bild 2). Dieser Baustein erlaubt es auch, nachträglich Schaltungen, die nicht für den Batteriebetrieb entwickelt wurden, auf diesen umzustellen.



Pin Functions

Timer mit MOSFET-Treiber

Das Blockdiagramm zeigt Bild 3. Ein Oszillator samt Teiler bestimmen den Takt, der Decoder übernimmt zwei Aufgaben: Nach dem Einschalten wird der Wert des an Pin DELAY/M_DRV angeschlossenen externen Widerstandes

ermittelt und so die Laufzeit des Timers in weiten Grenzen festgelegt. Anschließend wird der Widerstand per Multiplexer abgekoppelt, und Pin DELAY/M_DRV dient fortan als digitaler Eingang, um einen Taster für einen manuellen Reset zu erkennen. Der Block LOGIC CONTROL in Bild 3 enthält die benötigte Prozesssteuerung: Pin EN wird für den Timerbetrieb fest auf V_{DD} gelegt. Im One-Shot-Modus durchläuft der Timer nur einen Zyklus und hält danach an. Pin DRV steuert das Gate eines P-Kanal MOSFETS an. Dieser Ausgang wird LOW, sobald der Timer abgelaufen ist und die Schaltung mit Strom versorgt werden soll. Dann wird der P-Kanal MOSFET leitend und der Mikrocontroller kann seinen Job verrichten. Er signalisiert via Pin DONE mit einer Änderung von LOW zu HIGH, dass er seine Aufgabe erfüllt hat und bis zum nächsten Zyklus auf Spannung verzichten kann. Pin DRV wird also HIGH, der P-Kanal-MOSFET sperrt. Die Schaltung illustriert Bild 4, dem Datenblatt entnommen: Je nach Spannungsversorgung kann der hier vorhandene IC zum Power Management links im Bild entfallen. R_{EXT} bestimmt - wie beschrieben - die Laufzeit des Timers und wird in Prototypenschaltungen meist durch ein Potentiometer ersetzt. Die Zykluszeit ist flexibel zwischen 100 ms und 7200 Sekunden – letzteres entspricht zwei Stunden - justierbar. Im Datenblatt gibt es zur Berechnung des Widerstandes eine Formel und eine Tabelle für populäre Zykluszeiten. Was man unbedingt beachten muss: Der zulässige Spannungsbereich beträgt 1,8 Volt bis 5,5 Volt! Höhere Spannungen würden den Timer zer-

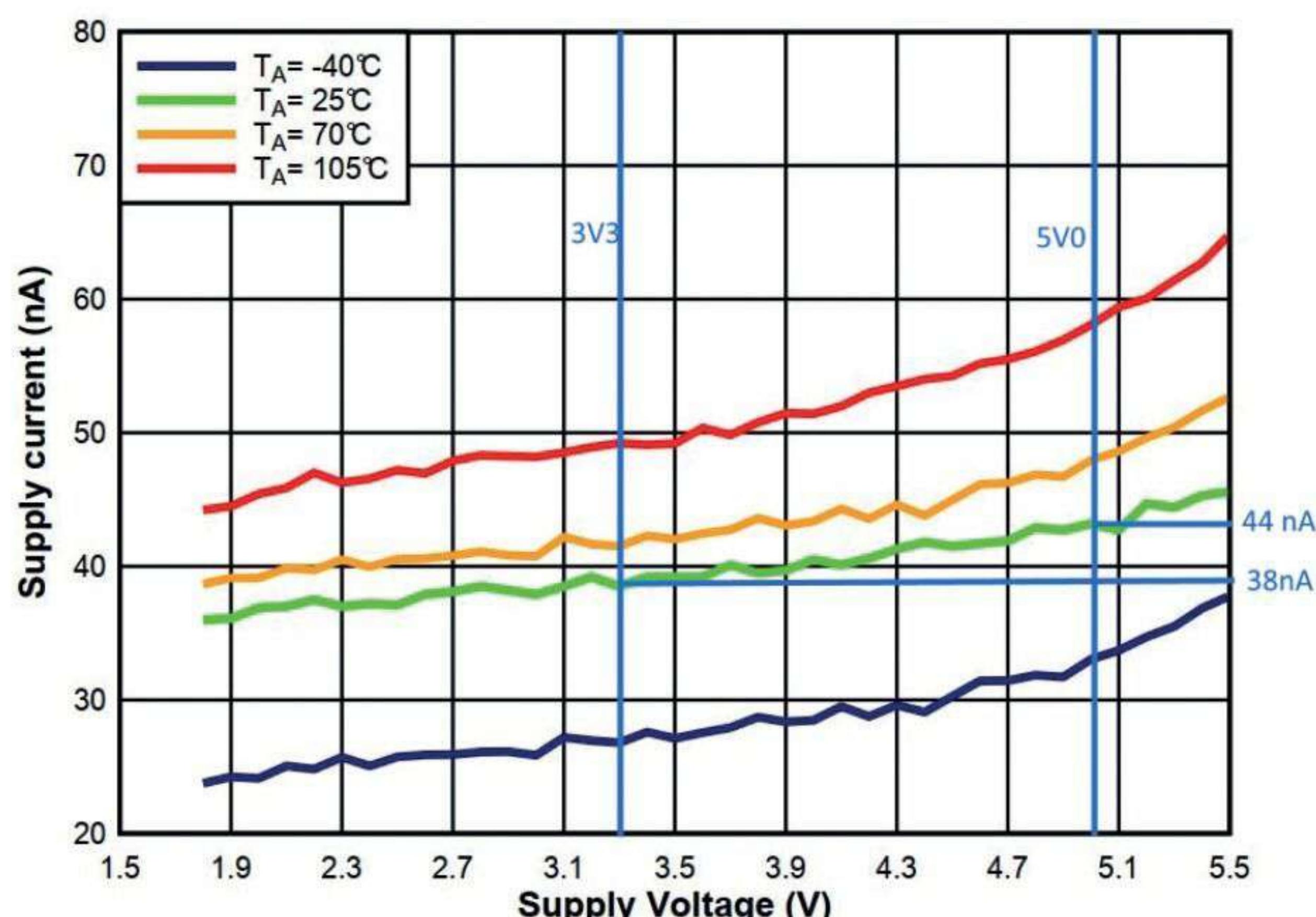


Bild 2: Das Diagramm zeigt die Stromaufnahme des Timers anhand der Betriebsspannung. Quelle: Datenblatt.

stören. Und: Der DONE-Eingang darf nicht floaten! Wird DONE nicht benutzt und mit GND verbunden, reagiert der Timer so: Der Timer läuft ab, DRV wird LOW und bleibt es 50 ms vor Ablauf der Zykluszeit. Dann wird DRV für diese 50 ms HIGH und ein neuer Zyklus beginnt. Im Oszilloskopogramm von DRV zeigt sich ein Dauerndes LOW, unterbrochen von 50 ms Highpegel. Der Normalfall ist jedoch, dass der Mikrocontroller via DONE meldet, dass sein Job getan ist.

Bauform: SMD

Die Bauform, die der Hersteller Texas Instruments für seinen Timer gewählt hat, ist ein sechspoliges SMD-Gehäuse mit der Bezeichnung SOT23. Damit der IC beim Zusammenstecken für Probeschaltungen handhabbar wird, wurde er auf eine kleine Lochrasterplatine bugsiert und mit dünnem Kupferdraht verlötet (Bild 5). Via übliche Pinheader sind die Anschlüsse für Dupont-Kabel (Jumperkabel) erreichbar. Weil es praktischer ist, wurde das kleine Breakout-Board mit doppelreihigen Anschlüssen ausgestattet, so ist jeder IC-Pin für zwei Kabel erreichbar. Das erleichtert oft die Verkabelung.

Ein Testaufbau bestand aus einem Arduino Nano (mit 5 V betrieben), einer LED als Last und einem P-Kanal-MOSFET mit der Bezeichnung Si2329DS. Dieser MOSFET-Transistor schaltet bei so geringen Spannungen von 3,3 und

5 Volt voll durch. Der maximale Spannungsunterschied zwischen Drain und Source darf 8 Volt nicht überschreiten. Das ist kein Problem, sofern die Batteriespannung nicht über 6 Volt ($4 \times 1,5$ -Volt-Zellen) festgelegt ist.

Rechenstunde

Lassen Sie uns einmal rechnen: Zunächst ohne Timer. Der Arduino Nano benötigt etwa 40 mA Strom. Auf 10 Minuten gerechnet sind das $40 \text{ mA} \times 600 \text{ Sekunden} = 24000 \text{ mA/s}$. Nun die Rechnung mit Timer: Die Zykluszeit soll weiterhin 10 Minuten betragen und der Prozessor soll 10 Sekunden einen Job verrichten. Dann addiert sich das so: $40 \text{ mA} \times 10 \text{ Sekunden} = 400 \text{ mA/s}$ plus $0,000044 \text{ mA} \times 560 \text{ Sekunden} = 0,02464 \text{ mA/s}$. Beide zusammen sind also $400 \text{ mA/s} + 0,02464 \text{ mA/s} = 400,02464 \text{ mA/s}$. Man sieht, dass der Betrieb des Timers kaum ins Gewicht fällt und der große Teil des Stromverbrauches in den zehn aktiven Sekunden stattfindet. In unserem Beispiel reduziert sich der Verbrauch von 24000 mA/s auf nur etwas über 400 mA/s. Noch beachtlicher wird die Differenz bei längeren Zykluszeiten des Timers.

Die Brüder

Texas Instruments hat für ähnliche Anwendungen einige Brüder des TPL5110 erschaffen. Der TPL5111 beispielsweise unterscheidet sich lediglich durch eine invertierte

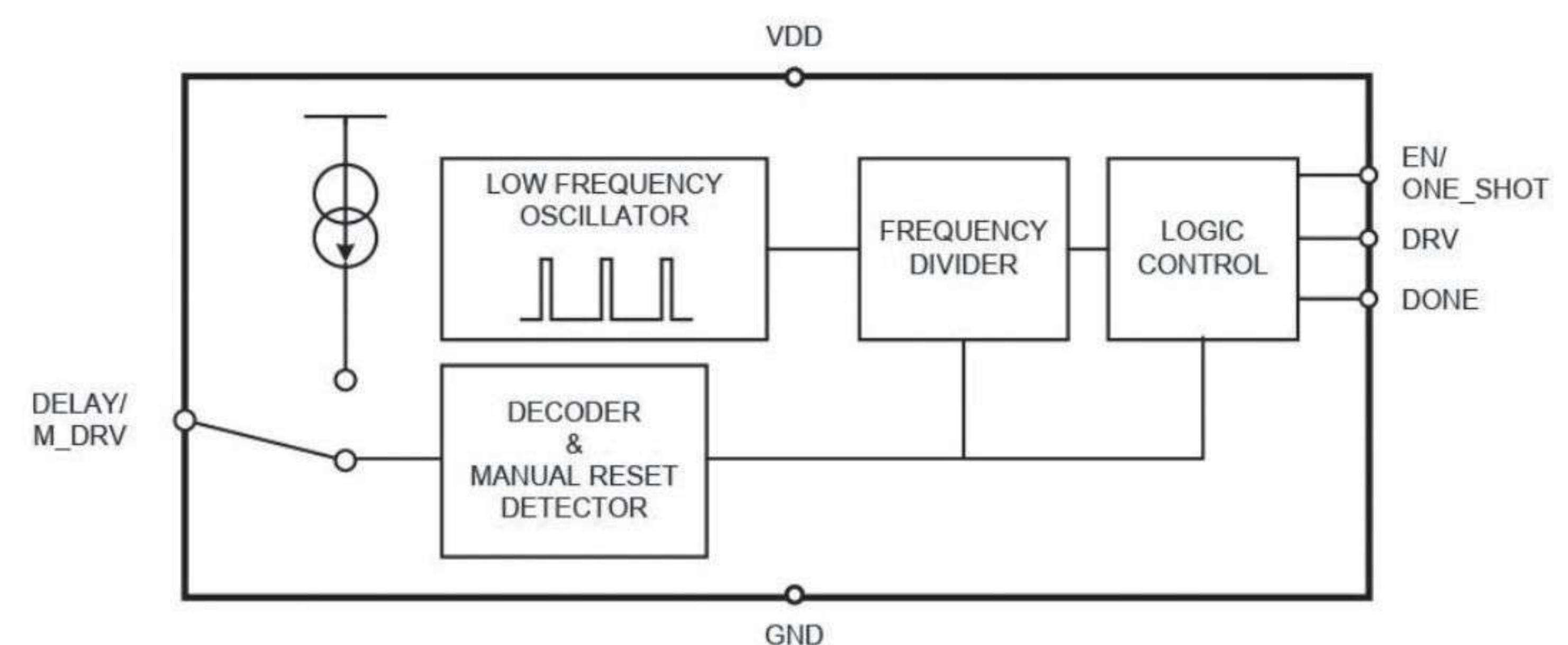


Bild 3: Das Blockdiagramm des Timers. Quelle: Datenblatt.

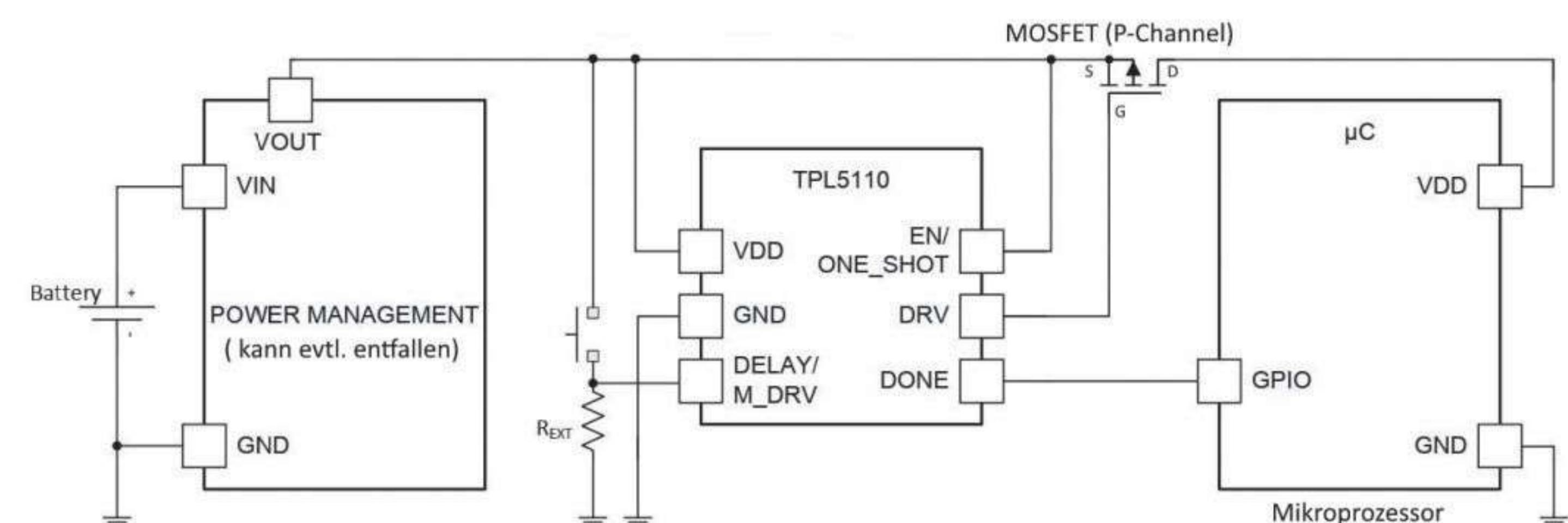


Bild 4: Prinzipielle Anwendung des Nano-Timers. Quelle: Datenblatt.

Ausgabe des DRV-Pins: Während beim TPL5110 Pin DRV, um die Spannung einzuschalten, LOW wird, schaltet der TPL5111 DRV auf HIGH. Mit diesem Signal ist es beispielsweise möglich, den Enable-Eingang eines Low-Drop-Reglers zu aktivieren und der Last Spannung zu geben.

Dann gibt es den TPL5010: Dieser verzichtet auf einen MOSFET-Treiber und verfügt stattdessen über einen Ultra-Low-Power-Watchdog. Der TPL5010 übernimmt die Funktionalität eines integrierten Timers des Prozessors. Der Mikrocontroller kann dann in einen wesentlich stromsparenderen Modus betrieben werden – bei ausgeschaltetem internem Timer – und erst nach einem Interrupt durch den TPL5010 zurück in den

aktiven Modus schalten. Durch Stromeinsparungen von fast zwei Größenordnungen ermöglicht der TPL5010 die Verwendung von deutlich kleineren Batterien für Interrupt-gesteuerte Anwendungen. Einige Normen (wie EN50271, Elektrische Geräte für die Detektion und Messung von brennbaren Gasen, giftigen Gasen oder Sauerstoff) erfordern die Implementierung eines Watchdogs, der TPL5010 realisiert diese Watchdog-Funktion fast ohne zusätzlichen Stromverbrauch.

Literatur/Verweise:

- [1] Datenblatt TPL5110: <https://www.ti.com/product/TPL5110>

Michael Wöste, DL1DMW

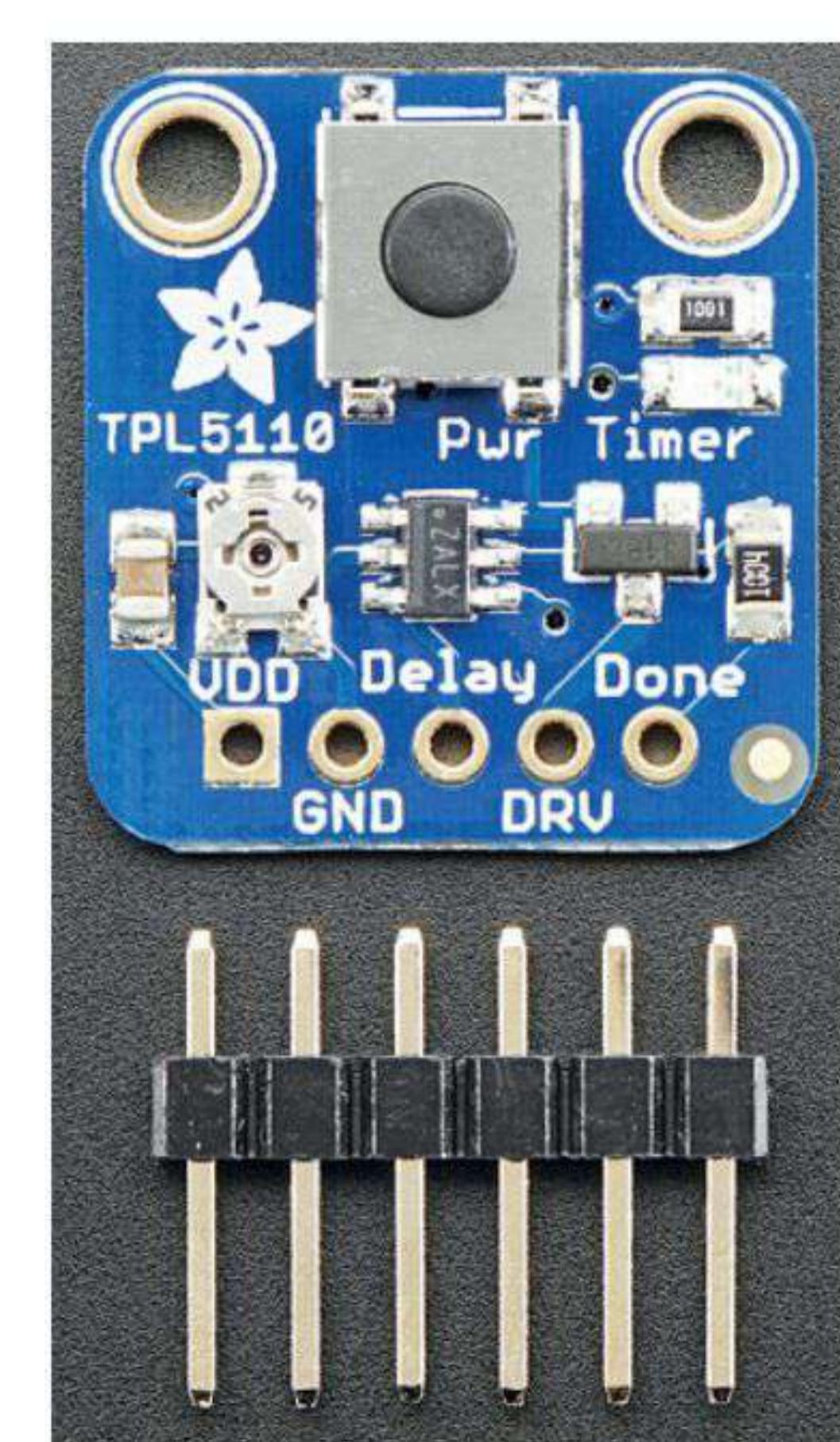


Bild 5: DIY Breakout Board für Prototypen.

Bild 6:
Von Adafruit gibt es ein fertiges Breakout Board für den TPL5110.

Die DX-LabSuite besteht aus einem Paket von Windows-Applikationen, die zusammen genommen das Anforderungsprofil eines Funkamateurs fast vollständig abdecken. DX-Lab steuert das Funkgerät, sorgt für das Logbuch, schaut online u.a. bei QRZ.COM vorbei, kümmert sich um die QSL-Karten und zeigt eine Weltkarte. Die Software achtet auf die Erfüllung allerlei Diplome und verfügt über eine integrierte HF-Ausbreitungsvorhersage. Einige der Module sind kostenpflichtig, andere kostenfrei. Zu letzten Modulen – die unabhängig von der DX-LabSuite läuft – ist der Spot Collector. Dieses Programm bietet Einblick gleichzeitig in bis zu vier DX-Cluster, bietet den Zugriff auf den IRC-Kanal von DX-Summit und empfängt – falls vorhanden – DX-Meldungen über einen lokalen TNC (Terminal Node Controller) oder über die AGW Soundcard Packet Engine.

Spots der gleichen Station auf der gleichen Frequenz und Betriebsart werden in einem einzigen Spot-Datenbankeintrag zusammengefasst und im Hauptfenster (Bild 1) angezeigt. Spot Collector identifiziert nicht nur benötigte (also zuvor nicht geloggte) Stationen, sondern deckt auch die Betriebsmuster einer DX-Station auf, damit die YL bzw. der Operator zur richtigen Zeit auf der richtigen Frequenz hört. Die Software ist auch dazu geeignet, die Öffnung von Sekundärbändern zu entdecken, insbesondere wenn DX-Stationen und Spotting-Stationen auf der Weltkarte von DXView angezeigt werden, einem anderen Modul der DX-LabSuite. Dem aufmerksamen Operator werden Einträge aus vermeintlich „geschlossenen“ Bändern jedoch in der Tabelle auffallen.

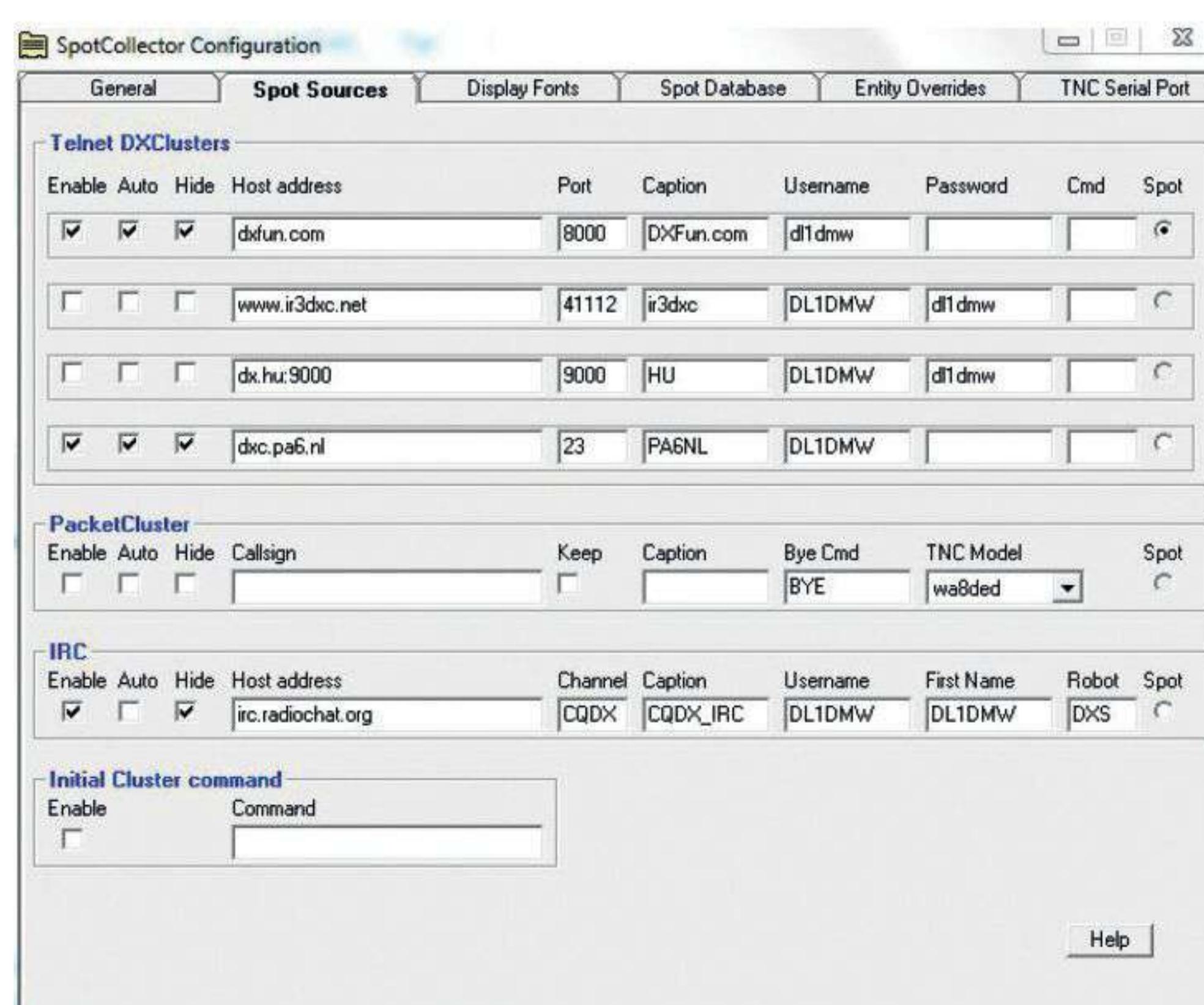


Bild 2: Im Konfigurationsdialog lassen sich bis zu vier Telnet-DX-Cluster als Datenquelle festlegen. Hinzu kommt ein IRC-Kanal und – wenn vorhanden – ein lokaler TNC via RS232.

Einblick in die Bänder DX-Lab Spot Collector

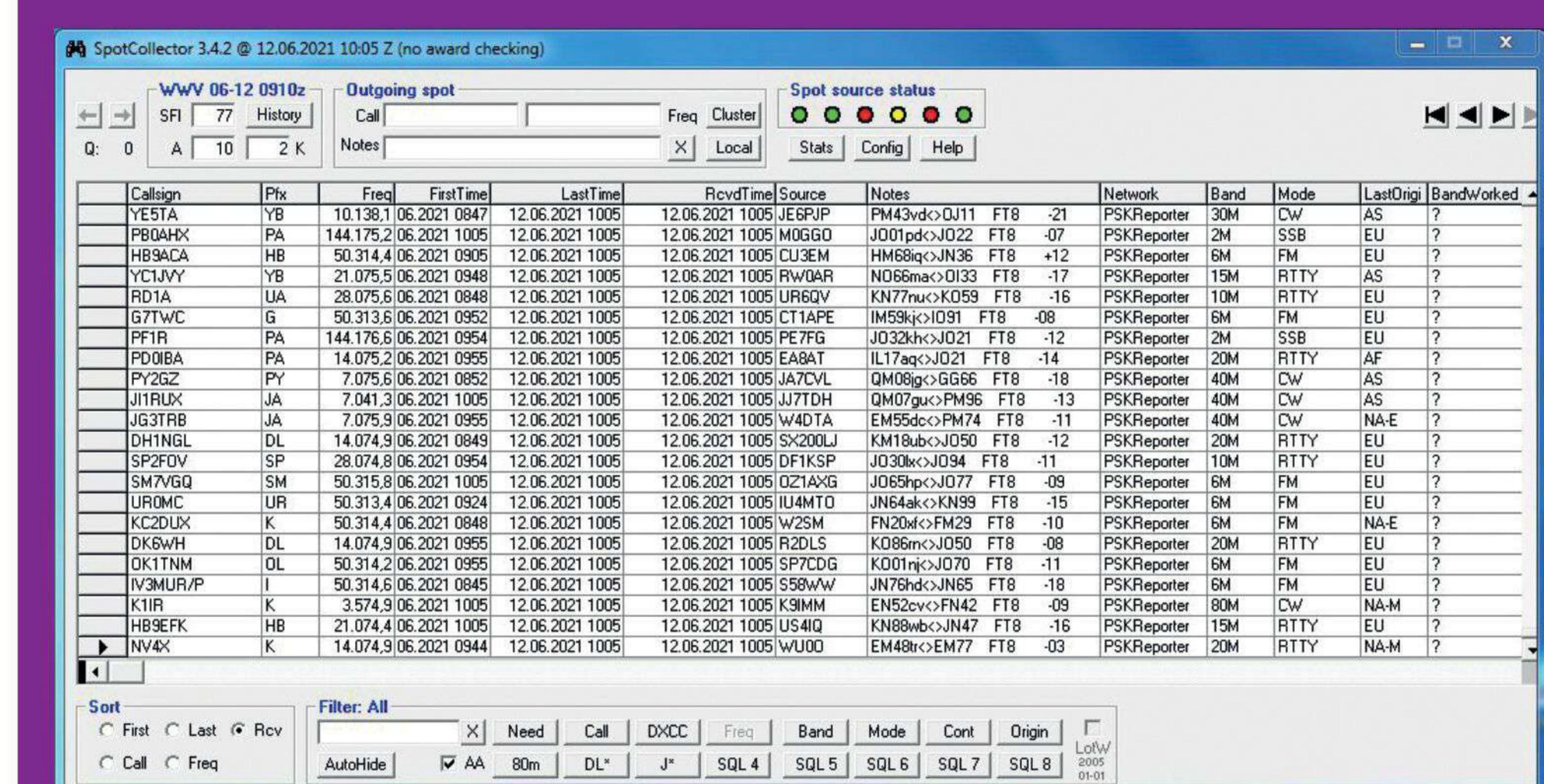


Bild 1: Der Spot Collector ist Teil der DX-LabSuite, läuft aber auch solo.

Der Spot Collector ist Teil der DX-LabSuite, läuft aber auch solo und ist kostenlos. Der Vorteil gegenüber einem Web-Cluster: Die Software zapft mehrere Quellen an und verfügt über flexible Filterfunktionen.

Das Windows-Programm verfügt unterhalb der Tabelle über eine Sortier- und Filterfunktion. Letztere ist sehr flexibel: Spots lassen sich nach Rufzeichen, Amateurfunkbändern,

dem Betriebsmodus, nach Kontinenten und mehr filtern. Ein besonderes Merkmal stellt die Möglichkeit dar, Filterbedingungen in Form von SQL-Befehlen nach definierter Syntax selbst zu formulieren (Bild 4) und den Button, über den man das Filter aufruft, selbst zu benennen. Auf diese Art lassen sich beliebig komplexe Filter definieren.

Während des Empfangs von Spots via Internet-DX-Cluster berechnet der Spot Collector Statistiken nach Band und Kontinent (Bild 5). Es informiert über die wichtigsten Werte der aktuellen Kurzwellenausbreitung mit SFI-, A- und dem K-Wert. Läuft das Programm einen Monat durch (oder wird jeden Tag gestartet), füllt sich

mit jedem Tag die grafische Statistik der drei solaren und geomagnetischen Parameter.

Spot Collector ist zwar allein lauffähig, entwickelt jedoch weitere Fähigkeiten nur in Verbindung mit anderen Modulen: Die Software stellt aktive DX-Stationen auf der Weltkarte von DXView dar, liefert Frequenz und Betriebsart an den Commander für One-Click-QSY und initialisiert beide VFOs für Split-Betrieb, falls das erforderlich sein sollte. Die Software liefert den solaren Flux (SFI) und den geomagnetischen K-Index an das PropView-Modul.

Der Spot Collector durfte nicht auf die virtuelle DVD („commercial use is strictly forbidden“) und so

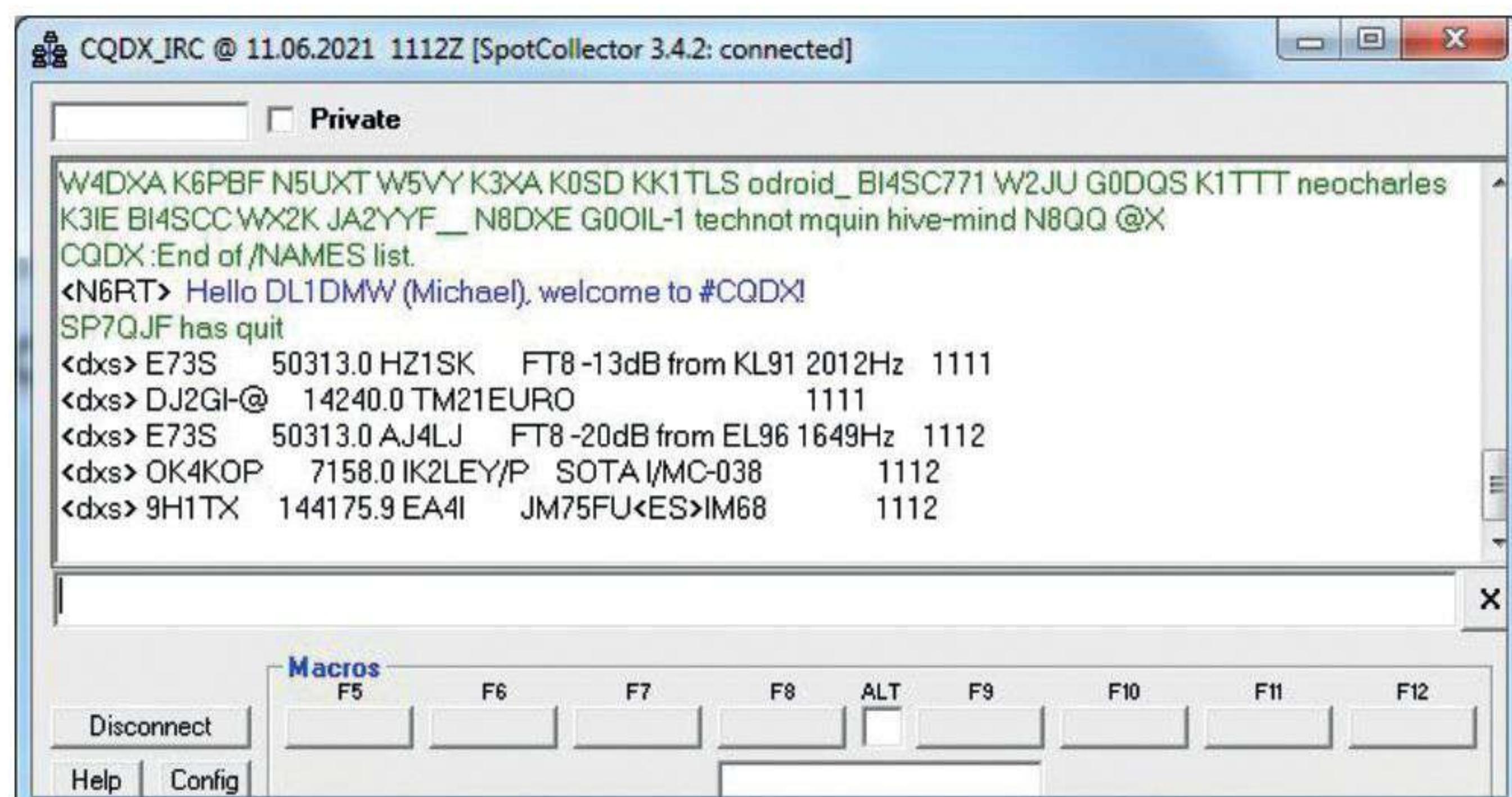


Bild 3: Zu jeder Datenquelle gibt es ein Terminalfenster. Es zeigt u.a. den Aufbau der Kommunikation und eingehende Daten. In der Konfiguration lassen sich diese Fenster ausschalten und die Verbindung zu den Quellen automatisch herstellen.

laden Sie die Datei bitte von [2]. Folgen Sie den Schritten der Installationsanleitung auf der Webseite. Die Datei zum Download finden Sie bei Schritt drei.

Literatur/Verweise:

[1] DX-Lab Spot Collector: <https://www.dxlabsuite.com/spotcollector/>

[2] Spot Collector Download: <https://www.dxlabsuite.com/spotcollector/download.htm>
[3] Hilfetext SQL-Beispiele: <https://www.dxlabsuite.com/spotcollector/Help/SpotDatabase.htm>

Michael Wöste, DL1DMW

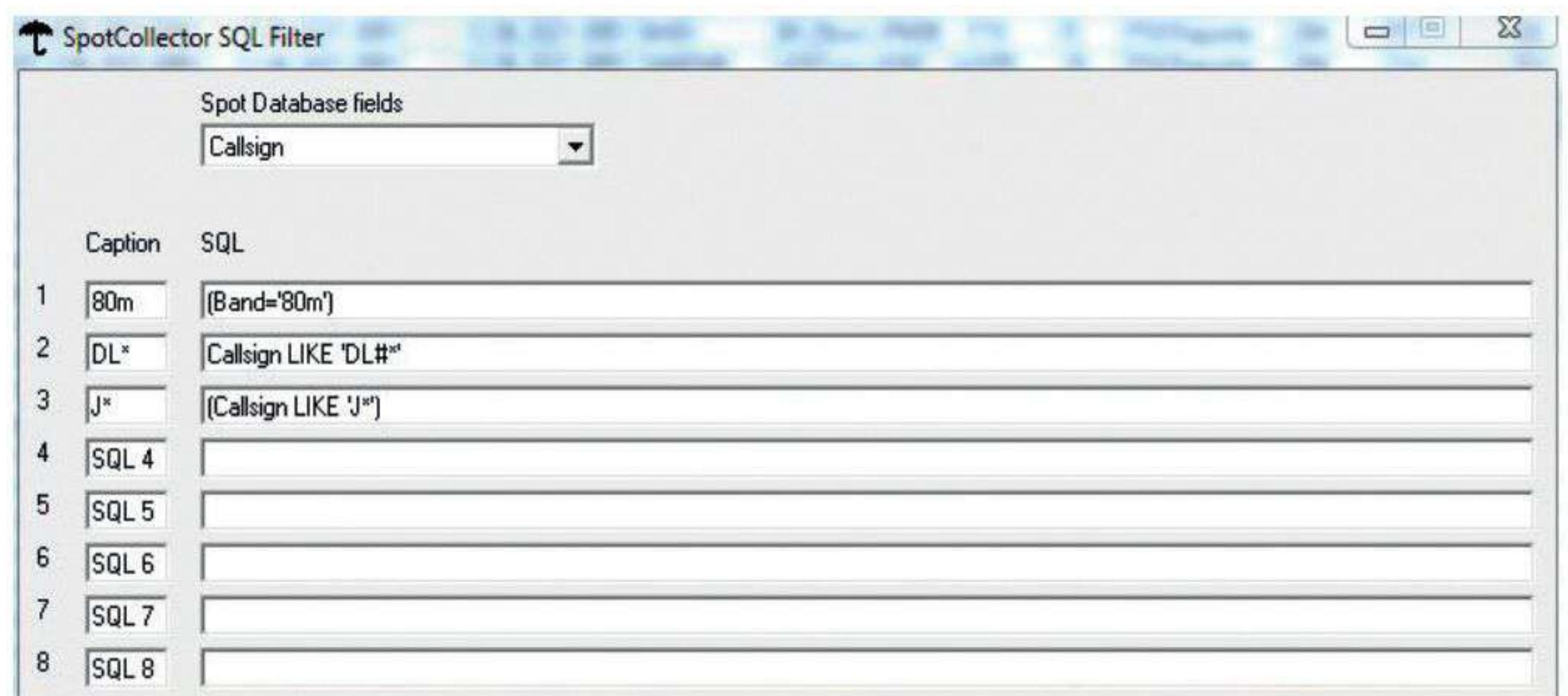


Bild 4: Mit einem STRG+linker Maustaste auf einen SQL-Filterbutton erscheint das Eingabemenü für QSL-Befehle. Es lassen sich beliebige Filterbedingungen zusammenstellen. Mehr Info: [3].

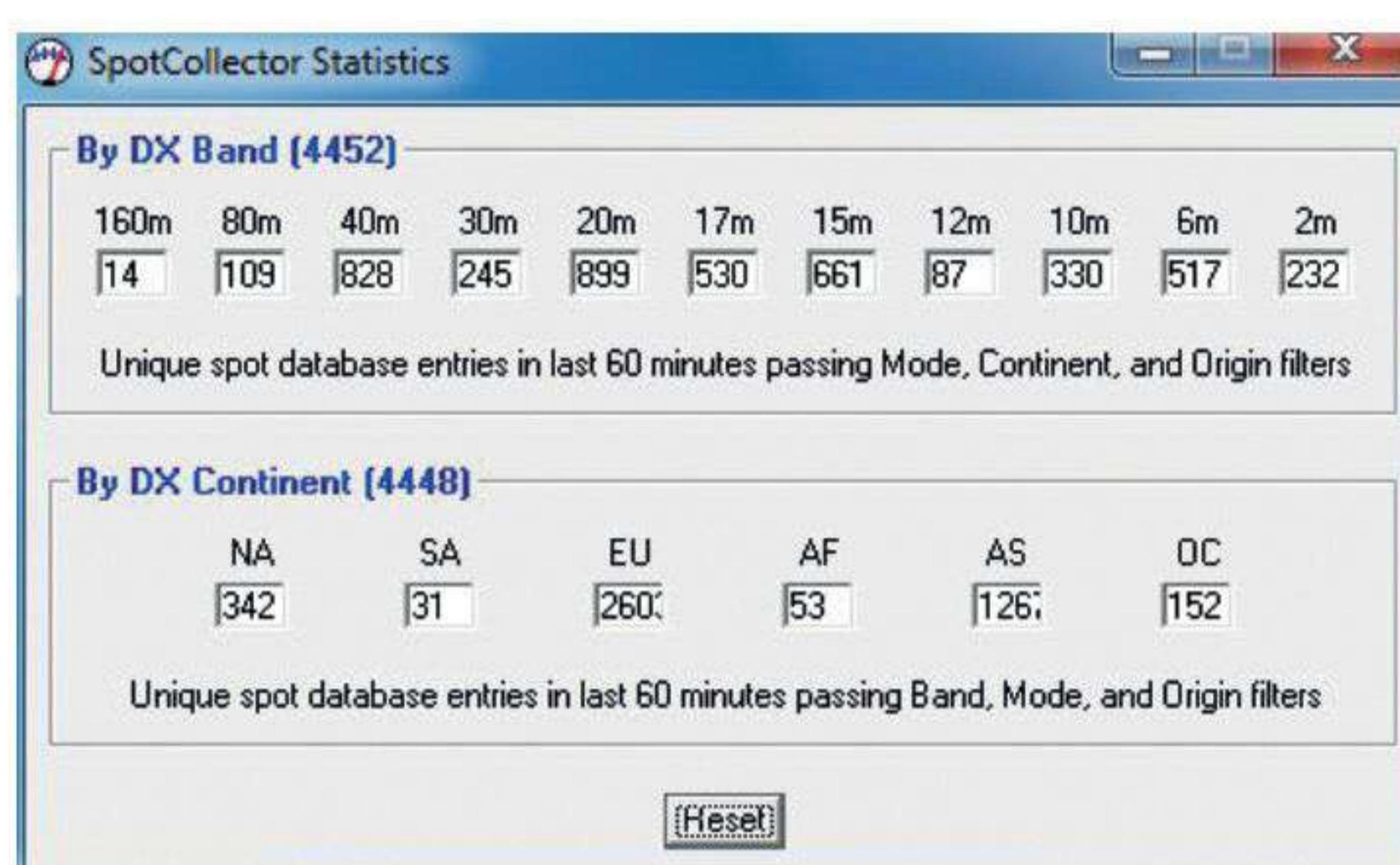


Bild 5: 60-Minuten-Statistik an einem Samstagmorgen. Es herrscht reger Funkbetrieb.

Funktionsgenerator mit ESP32: Sinus, Rechteck, Dreieck

Zum Basteln benötigt man ab und zu einen Funktionsgenerator. Was tun, wenn keiner zur Hand ist? Mit einem ESP32 – an anderer Stelle dieses Heftes vorgestellt – hat man in 5 Minuten einen NF-Funktionsgenerator mit Sinusschwingungen bis 250 kHz oder 20 kHz für Rechteck- und Dreiecksignale mit einer Spannung von Spitze-Spitze 3,3 Volt. Wer höhere Spannungen benötigt, schaltet einen NF-Verstärker nach.

Im Internet bin ich auf der Suche nach einer schnellen Lösung für einen Funktionsgenerator zuerst bei [1-3] fündig geworden. Gerald Lechner beschreibt im ersten Teil seines Blogbeitrags eine Software für den ESP32, die über den seriellen Monitor der Arduino IDE bedient wird. Genau das benötigte ich, eine Software, die man schnell auf den ESP32 aufspielt, sie nutzt und anschließend den ESP32 für andere Zwecke verwendet. Der Autor des Programms geht in seiner Beschreibung in Teil 2 und Teil 3 jedoch weiter: Die Bedienung wird auf Taster umgestellt und die Hardware in ein selbstgedrucktes Gehäuse



Bild 1: Ausgabe des Sinussignals mit 2700 Hz, oben das Nutzsignal (GPIO26/DAC2), unten das auf GPIO25 ausgegebene Signal.

integriert. Soweit wollte ich nicht gehen, schließlich ging es mir nur

um eine schnelle Lösung für ein Sinussignal.

Sein ESP32-Programm, das man mit der Arduino IDE lädt und kompiliert, habe ich nur geringfügig auf deutschsprachige Befehle umgestellt und die Eingabeaufforderung, die man im seriellen Monitor lesen kann, so gestaltet, dass nach längerer Nichtbenutzung keine Fragen zur Bedienung bzw. zu den Kommandos auftreten. Aus Zeitgründen greift der Programmierer direkt auf die Register des Timers, des 8-Bit-DAC und der I2S-Schnittstelle zu. Gibt der Benutzer eine Wunschfrequenz vor, etwa 10.000 Hertz, antwortet die Software mit einer angenäherten Frequenz, die von dieser meist nur um wenige Hertz abweicht (hier: 10001.25Hz). Die leichte Differenz ist in der Hardware des ESP32 begründet. Die Software versucht, die reale Frequenz möglichst der Wunschfrequenz anzunähern – das funktioniert sehr gut.

Am ESP32 nimmt man das Signal an Pin GPIO26 ab. Das ist der DAC2-Ausgang. Auf GPIO25



Bild 2: Ausgabe eines Rechtecksignals mit selber Frequenz, auch hier oben das Nutzsignal auf dem ESP32 Pin GPIO26/DAC2.

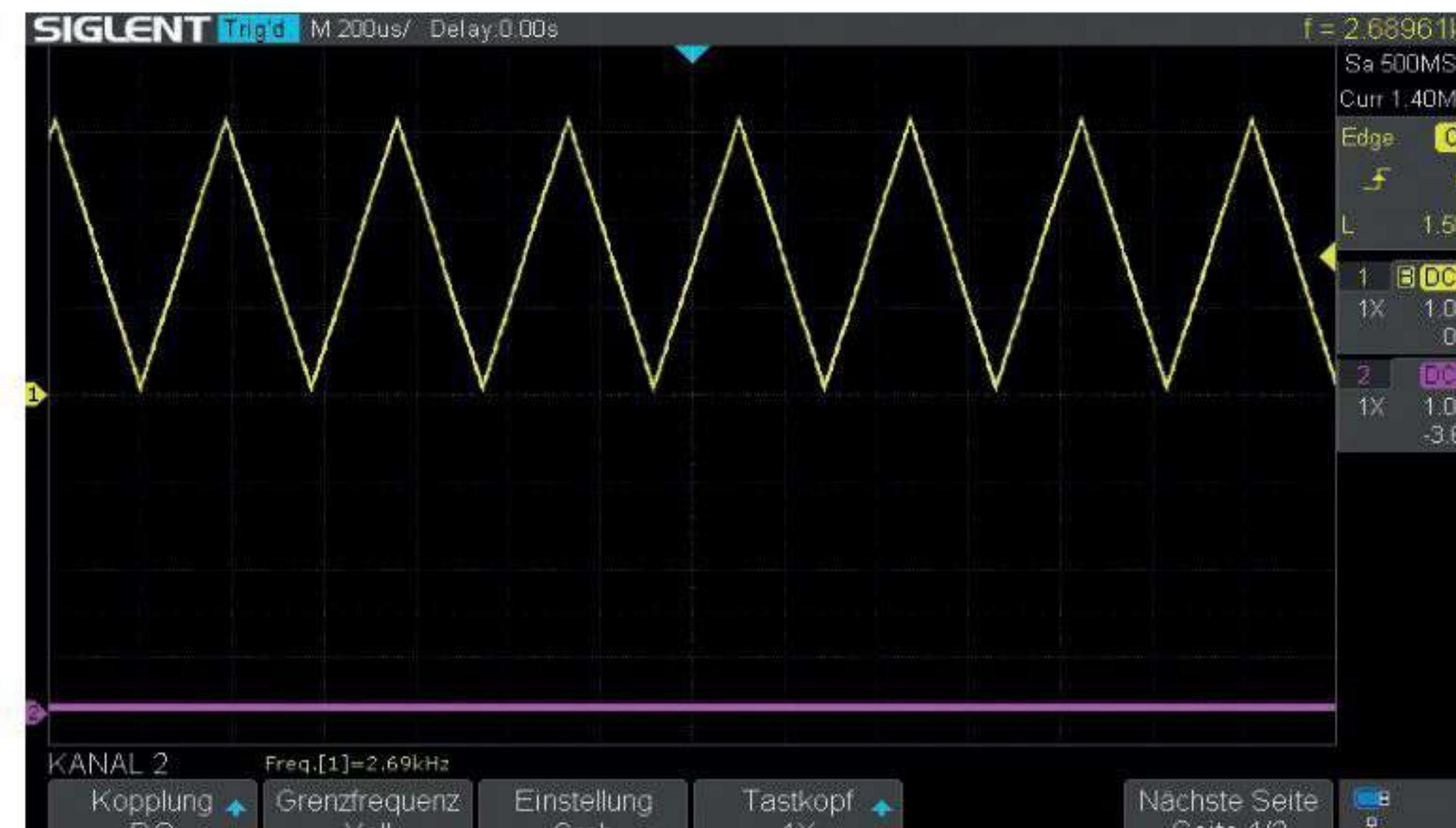


Bild 3: Dreiecksignal mit 2700 Hz.

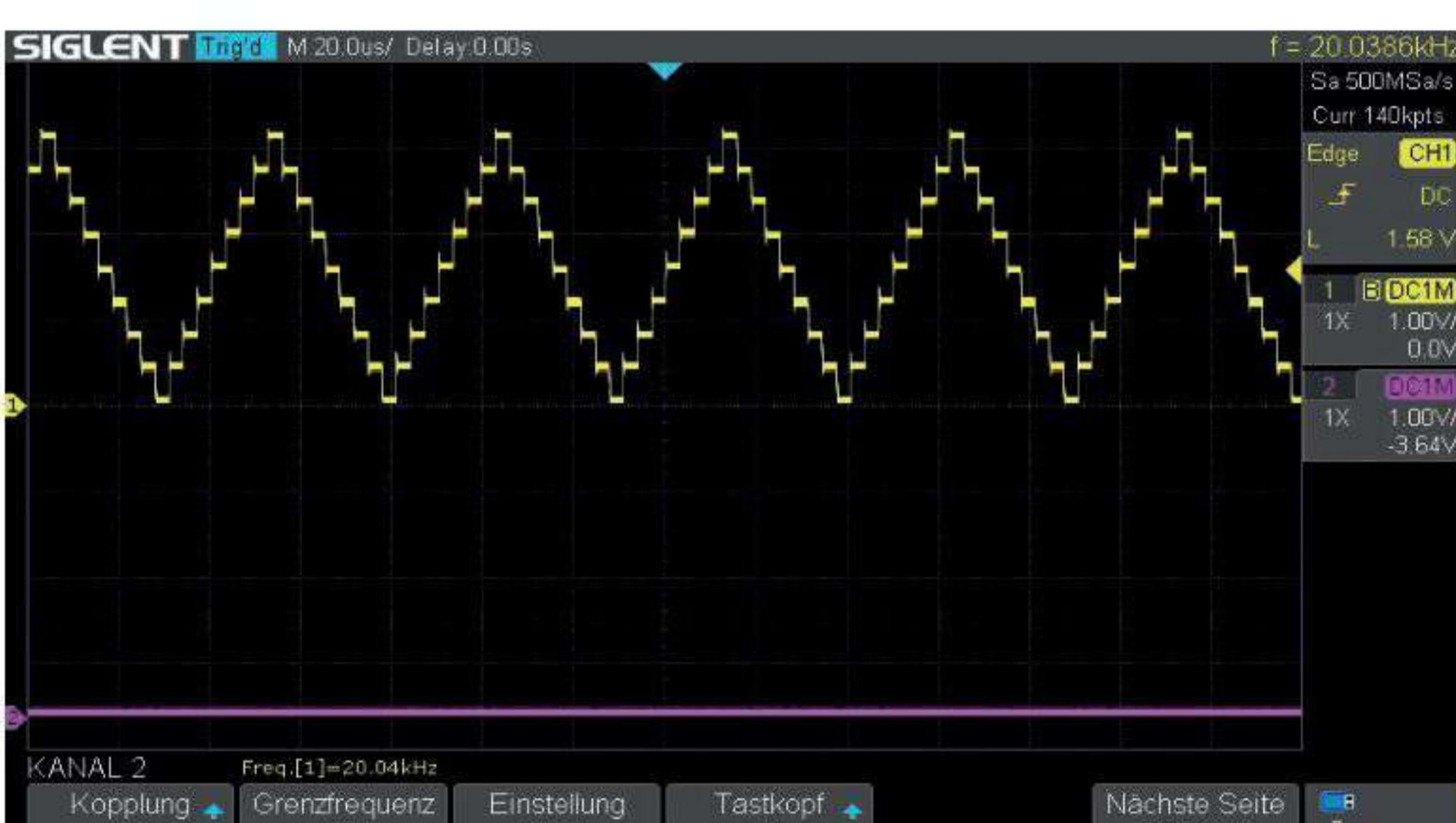


Bild 4: Bei einer Frequenz von 20 kHz weist das Signal deutliche Treppenstufen auf, während der Sinus und das Rechteck bei 20 kHz einwandfrei sind.

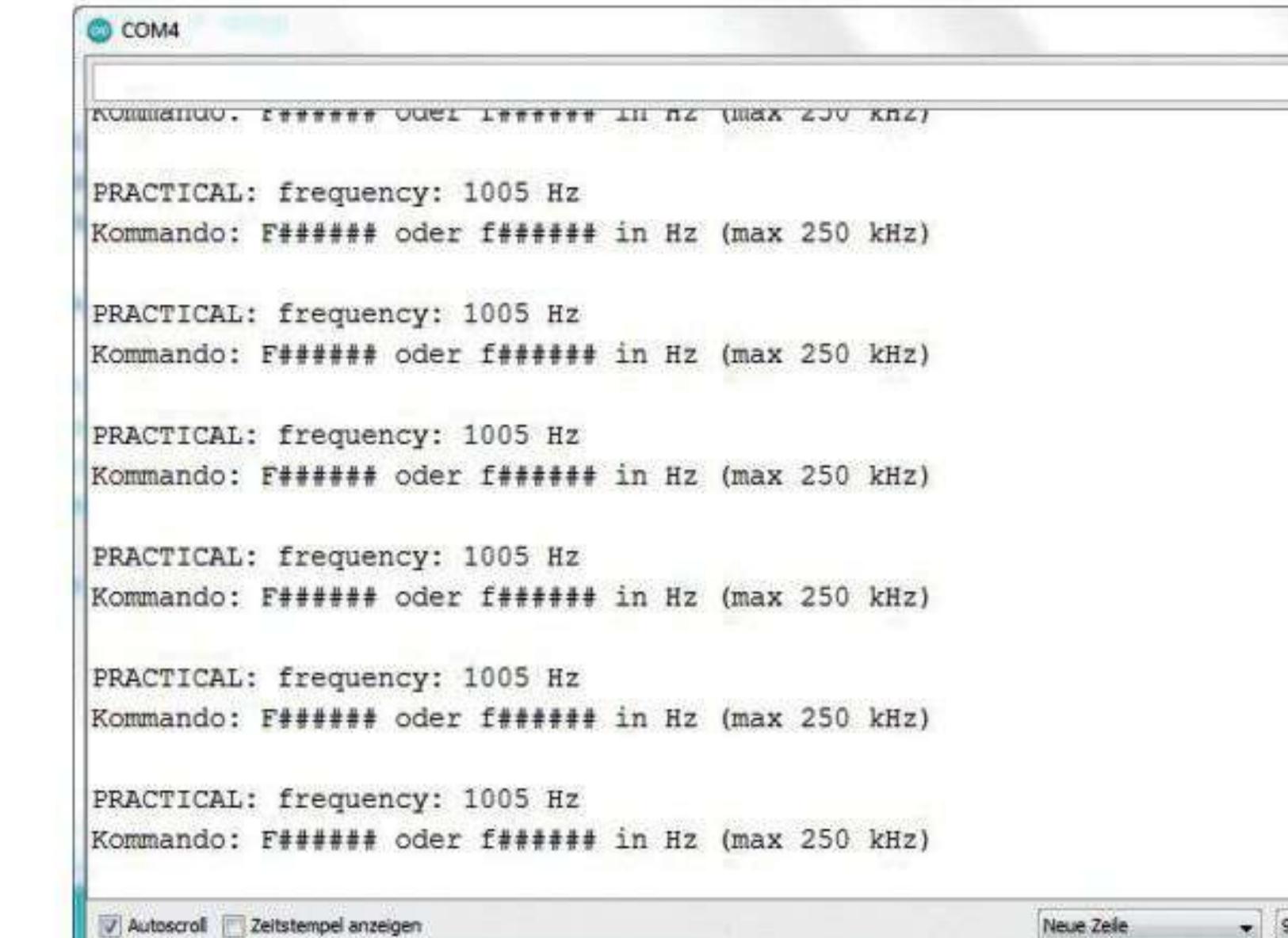


Bild 8

Befehl zur Einstellung der Frequenz ausgestattet. Der lautet

F##### oder f#####

und wird wie zuvor im seriellen Monitor eingetippt und gesendet. Beachten Sie, dass Sie hier sechs Ziffern angeben können. Eingaben über 250000 wirken aber nicht. Rechteck und Dreieck leistet dieser Sketch nicht. Das Signal können Sie an dem ESP32-Pin GPIO25/DAC1 oder auch bei GPIO26/DAC2 abnehmen. Nutzen Sie beide, ist es ein phasengleiches Stereo-Sinussignal. Bild 7 zeigt beide Signale bei 255 kHz und Bild 8 die Ausgabe des Sketches auf dem seriellen Monitor der Arduino IDE. Viel Spaß mit diesen beiden Programmen!

Literatur/Verweise:

- [1] ESP32_FuncGen1: <https://www.az-delivery.de/blogs/azdelivery-blog-fur-arduino-und-raspberry-pi/funktionsgenerator-mit-dem-esp32-teil1>
- [2] ESP32_FuncGen1: <https://www.az-delivery.de/blogs/azdelivery-blog-fur-arduino-und-raspberry-pi/funktionsgenerator-mit-dem-esp32-signalverstaerker-teil2>
- [3] ESP32_FuncGen1: <https://www.az-delivery.de/blogs/azdelivery-blog-fur-arduino-und-raspberry-pi/funktionsgenerator-mit-dem-esp32-display-und-gehaeu-se-teil-3>

- [4] ESP32_FuncGen2: <https://www.esp32.com/viewtopic.php?t=10321>
- [5] Sinus-Routine für ESP32_FuncGen2: <https://github.com/krzychb/dac-cosine>

Michael Wöste,
DL1DMW

Beispiele:

F1000 = Stellt die Frequenz auf 1000 Hz ein.

ms = Stellt die Ausgabe auf Sinus um.

MD = Ausgabe Dreiecksignal.

mr = Ausgabe Rechtecksignal.

T10 = Das Tastverhältnis wird auf 10% eingestellt.

... und nun Sinus bis 250 kHz

Es muss doch noch etwas schneller gehen, mag man denken, schließlich besitzt der ESP32 im DAC einen eigenen Sinusgenerator! Fündig wurde ich bei [4]. Helmut Weber hat mit Hilfe von anderer Seite [5] ein freies Beispielprogramm geschrieben, dass einen Sinus bis 250 kHz ausgibt. Das schafft eine PC-Soundkarte nicht! Ich habe es ESP32_FuncGen2 genannt und es für die Arduino IDE portiert. Weil man die Frequenz nicht einstellen konnte – sie war fest vorgegeben – kam noch die Abfrage der seriellen Schnittstelle hinzu und der Sketch wurde mit einem simplen

Auf der virtuellen DVD zum Download finden Sie den modifizierten Arduino-Sketch ESP_FuncGen1.ino im Verzeichnis \Artikel\ESP32FuncGen\Arduino. Für eine detaillierte Beschreibung der internen Abläufe zur Erzeugung der Signale empfehle ich den Blogeintrag des Programmierers bei [1].

Statt der Großbuchstaben sind auch kleine Buchstaben zugelassen. Den seriellen Monitor stellt man auf 9600 Baud ein.

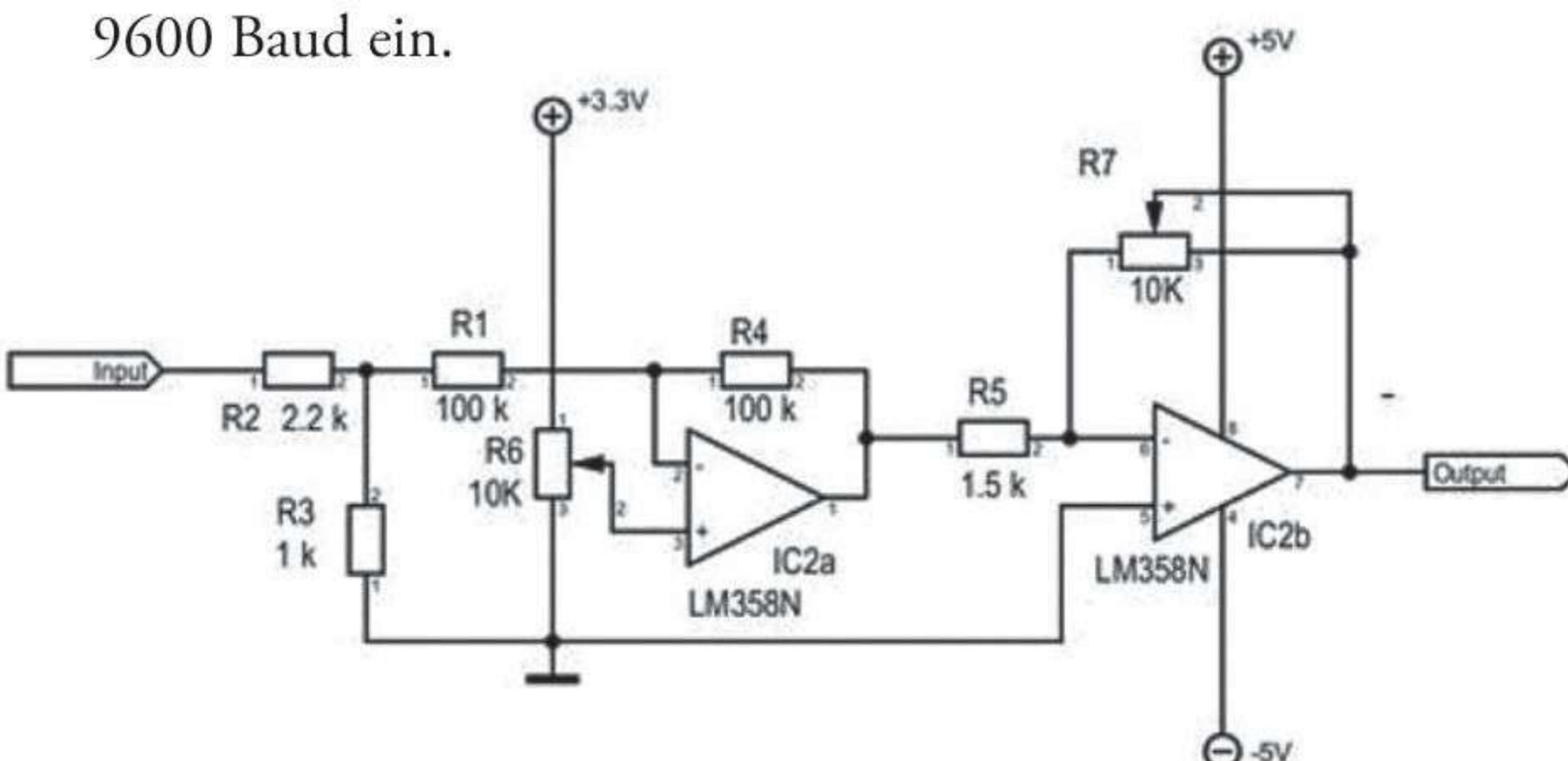


Bild 6: Ein NF-Verstärker mit LM358 erzeugt höhere Spannungen.
Quelle: [1-3].



Bild 7



Bild 1: Der ATU-100 wird mit nur einem Knopf bedient.

Angepasst: N7DCC-ATU-100 China-Nachbau

Der kleine Autotuner von David Fainitski, N7DCC, ist recht bekannt und beliebt. In den USA als Open Source - Open Hardware-Projekt entwickelt, liefern meist chinesische Quellen preiswerte Nachbauten. Wir betrachten ein Fertigerät, das über einen Webshop in Deutschland bezogen wurde.

Den kleinen ATU-100 im stabilen Aluminiumgehäuse mit dem Maßen 100*40*150 mm (B*H*T) gibt es als Fertigerät, als Bauatz mit und ohne Gehäuse und wahlweise mit 5 Abstimmgruppen (Kombination aus Induktivitäten und Kapazitäten) sowie in einer

erweiterten Version mit 7 Abstimmgruppen. Man bekommt Versionen mit klassischem LC-Display oder mit kompakten OLED-Displays. Die OLED-Displayfläche ist nur 25x17 mm klein, bietet Platz für vier Zeilen Text mit Angabe zur Leistung, Stehwellenverhältnis und



Bild 2: Die qualitativ ungenügenden PL-Buchsen wurden ausgetauscht.

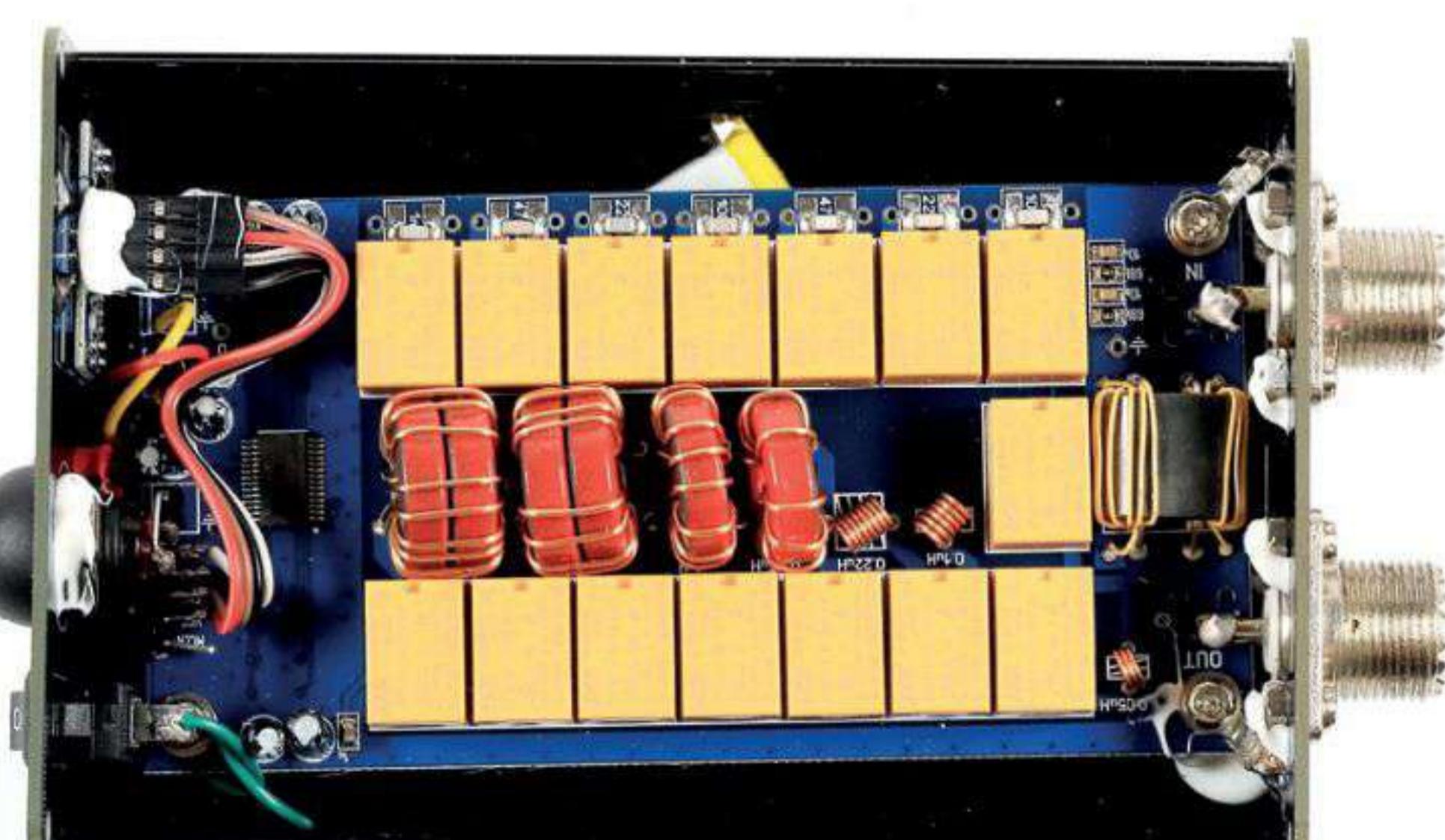


Bild 3: Die erweiterte Variante besitzt sieben Abstimmgruppen und ein breites Abstimmverhalten.



Betrieb

Ein Autotuner wird bei kleiner Leistung mit ca. 5 bis 10 Watt Leistung abgestimmt. Ich schalte den TRX dazu zuerst auf AM oder FM, drücke am ATU-100 den Tune-Knopf einmal und betätige danach die PTT-Taste. Nachdem der Tuner abgestimmt hat, stelle ich das Funkgerät auf die gewünschte Modulationsart (CW, SSB) ein und erhöhe danach die Leistung auf das gewünschte Maß. Getestet wurde der kleine und für den portablen Betrieb mit nur 370 g Gewicht bestens geeignete ATU-100 an verschiedenen, nur halbwegs abgestimmten Antennen. Stets gelang in nur ein bis drei Sekunden eine gute Anpassung. Eine spätere Abstimmung an derselben Antenne und Frequenz gelingt in wenigen hundert Millisekunden. An einer Mobilantenne HF-Explorer (die mit dem Kurzschlusskabel ähnlich wie die Outbacker) und Funkbetrieb im (geparkten) Auto entfällt die störende Längenabstimmung des oberen Antennenelementes – das erledigt der kleine Automatiktuner.

Fazit: Ein kompakter Autotuner, der in jeden SOTA-Wanderrucksack zur Funkausrustung passt und sicherlich viel Funkfreude bereiten wird. Falls ein Direktimport aus China erwogen wird, sei auf die seit Juli erhöhten Portokosten und auf die von Zoll erhobene deutsche Einfuhrumsatzsteuer hingewiesen, die manchen Import möglicherweise unattraktiv erscheinen lässt. Das ist zuvor zu prüfen.

Literatur/Verweise:

- [1] N7DCC-Github: <https://github.com/Dfinitski>
- [2] Shop in DL: https://www.kaufland.de/product/365764651/?search_value=ATU-100
- [3] Mehr Info zum AT-100: http://oe1iah.at/Hardware/AutoTuner_ATU100.shtml

Michael Wöste, DL1DMW

Bild 4: Beim Sendebetrieb informiert das OLED-Display über SWR, Leistung an der Antenne und die Effizienz der Antennenanlage.

Ein Timer-Tutorial mit Atmel ATmega328

DIGITAL-ANALOGWANDLER

mit Software

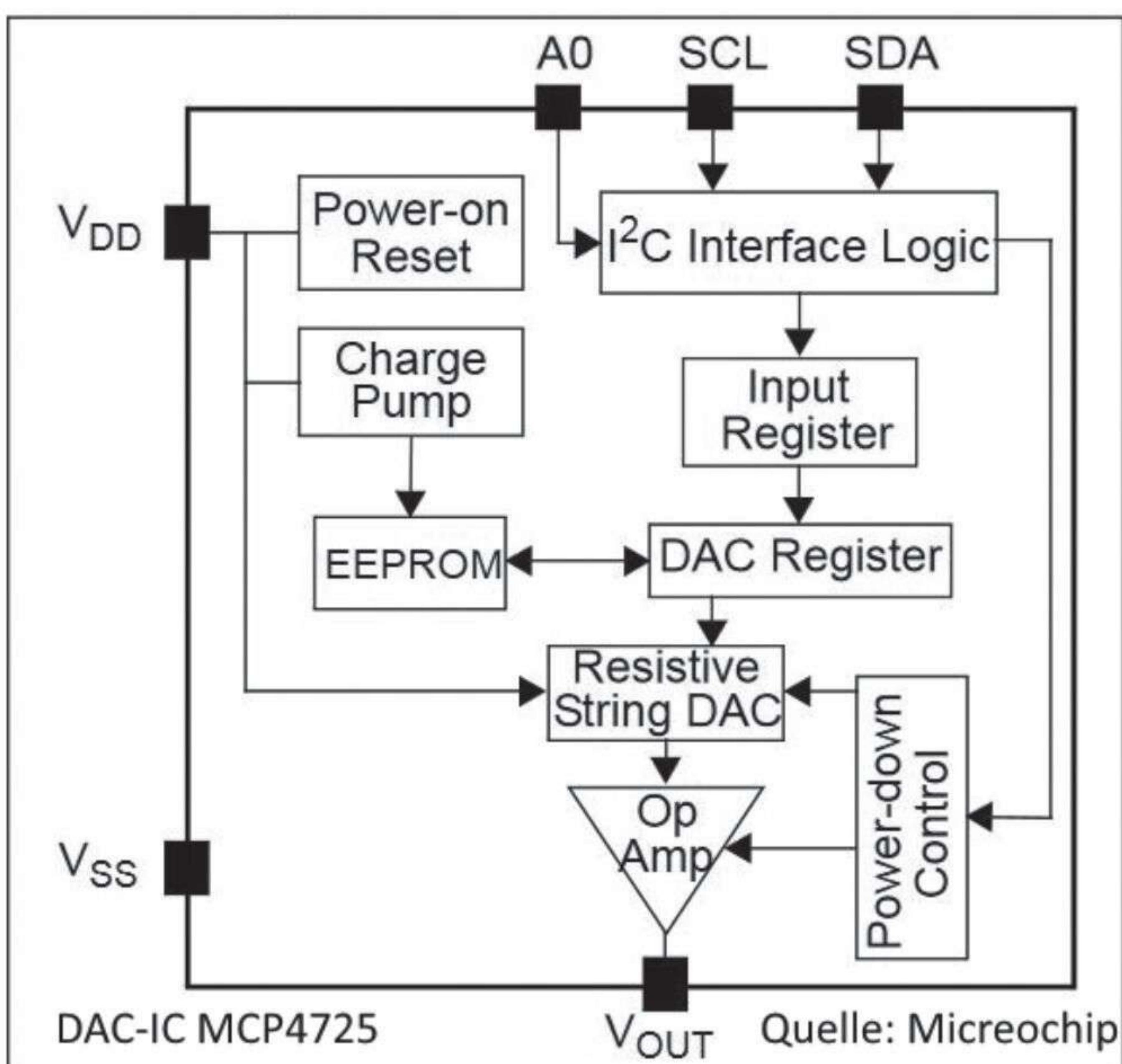


Bild 1: Funktionsschaltbild eines Digital-Analog-Konverters.

Die meisten Mikrocontroller haben keinen eingebauten Digital-Analog-Wandler (DAC = digital analog converter) und externe Wandler kosten Geld. Es gibt jedoch einen einfachen und preiswerten Trick, wie man einen analogen Ausgang simulieren kann. Dies ist ein kurzes Tutorial zur Erzeugung analoger Ausgangssignale mit einem preiswerten Mikrocontroller. Die analogen Signale werden durch Mittelwertbildung einer digitalen Pulsweitenmodulation (PWM) von einer der Zähler/Timer im Mikrocontroller erzeugt und das ohne spezielle Digital/Analog-Wandler. Wir stellen zunächst einige Aspekte des Zählers/Timers vor und betrachten, wie er zur Erzeugung eines Pulsweitenmodulationssignals verwendet werden kann. Danach werden wir das Funktionsprinzip auf einem AVR-Mikrocontroller implementieren, um einen analogen Ausgang zu simulieren und

schließlich eine einfache, kleine Funktionsgeneratorschaltung zu bauen.

D/A auf die schnelle Art

Die offensichtliche Lösung zur Erzeugung eines analogen Ausgangssignals in einem digitalen Gerät ist die Verwendung eines speziellen Digital-Analog-Wandlers (DAC), oft ein eigener IC, der extrem genau sein kann, wenn er eine sehr hohe Auflösung hat. Die kosten Geld und manchmal möchte man es stattdessen einfach auf die schnelle Art realisieren. Hier besteht der - zugegeben nicht neue - Trick darin, ein digitales Ausgangssignal mit Pulsweitenmodulation (PWM) zu verwenden und das Signal mit einem Tiefpassfilter zu mitteln, um das analoge Ausgangssignal – eine Gleichspannung – zu erzeugen. Im Vergleich zur Verwendung eines

Sie haben einen Mikrocontroller und wollen damit etwas Analoges steuern? Das ist eine häufige Aufgabenstellung, zu der es eine Reihe von Lösungen gibt, je nachdem, was genau Sie programmieren möchten.

Timer im ATmega328

Das zusammenfassende Datenblatt für den ATmega328 (der ersten Generation) weist darauf hin, dass auf jedem Chip drei unabhängige Zähler-/Timer-Schaltungen vorhanden sind. Da diese Schaltungen entweder für das Zählen von Ereignissen oder für das Timing von Ereignissen - d.h. das Zählen von Taktzyklen - verwendet werden können, werden sie oft als Zähler oder Timer bezeichnet, oder beides, je nach Kontext. Zwei der Schaltungen sind 8-Bit-Timer und der andere ist ein 16-Bit-Timer. Jeder Timer-Schaltung sind zwei PWM-Ausgänge zugeordnet, so dass insgesamt sechs PWM-Ausgänge auf dem Chip möglich sind. In einer neuere Version des Controllers mit der Bezeichnung ATmega328PB wurde die Zahl der 16-Bit-Timer auf drei erhöht.

Diese Technik ist – wie gesagt – nicht neu, sie ist einer der Standardtricks der Mikrocontroller-Programmier. Es gibt sogar eine Atmel Application Note zur Verwendung der PWM-Ausgänge als Analogausgänge. Wenn Sie diese noch nicht gesehen haben, gerade erst mit Mikrocontrollern anfangen zu programmieren oder die Application Notes undurchsichtig finden, könnte dieses kleine Tutorial für Sie hilfreich sein. Voraussetzung ist, dass Sie bereits einen Mikrocontroller-Platine wie einen Arduino UNO etc. besitzen und wissen, wie man Code kompiliert und auf einen Mikrocontroller lädt. Wenn Sie das mit einem Arduino UNO können, dann machen Sie hier mit!

Es wird wohl kaum einen gängigen Mikrocontroller geben, der nicht über einen oder mehrere Zähler/Timer verfügt. Da die Funktionsweise dieser Timer bzw. Zähler zwischen den verschiedenen Mikrocontroller-Familien sehr ähnlich ist, sind die grundlegenden Techniken weitgehend plattformunabhängig. Dennoch: Genaue Registerdefinitionen und Fähigkeiten der Hardware variieren manchmal selbst zwischen Chips derselben Mikrocontroller-Familie. Konsultieren Sie das Datenblatt! Für unsere Beispiele werden wir mit dem bekannten Atmel ATmega328 arbeiten.

und „B“ als Bezeichnung für die beiden Ausgänge. Beim ATmega328 im DIP-28-Gehäuse befinden sich OC1A und OC1B an den Pins 15 bzw. 16 und nach der Nummerierung, wie sie Arduino verwendet, auf Arduino Pin 9 und 10.

Wie PWM funktioniert

Jetzt geht es an die Feinheiten der Ansteuerung des Zählers/Timers, und hier beginnt es interessant zu werden. Beginnen wir mit einem kurzen Überblick über die Pulswidtemodulation und wie der Timer funktioniert. Eine Zähler/Timer-Schaltung kann in vielen verschiedenen Modi verwendet werden, doch wir werden nur einen davon im Detail behandeln. Zunächst einmal ist der Zähler nicht mehr als eine Variable in einem Register (im Controller) die anzeigt, wie oft ein bestimmtes Ereignis stattgefunden hat. Sie können beispielsweise Gegenstände auf einem Fließband mittels einer Lichtschranke zählen. Ein Zählimpuls der Lichtschranke bewirkt, dass der Zähler um eins erhöht wird. Man kann den Zähler/Timer auch dazu verwenden, um ein periodisches Taktsignal zu zählen. In diesem Fall könnte man den Zähler einen Timer nennen. Das schnellste Signal, das wir zählen können, ist der interne Systemtakt des Controllers. Unser Chip hat das Glück, einen internen RC-Taktoszillator mit einer maximalen Rate von 8 MHz zu besitzen. Man könnte nun diesen internen 8-MHz-Takt verwenden, doch wir wollen mit dem Arduino UNO (oder ähnlichen) arbeiten und der wird durch einen externen 16-MHz-Quarz getaktet. Wir werden diesen 16-MHz-Takt verwenden und ihn in die Zähler/Timer-Schaltung schicken.

Beispiel: Arduino

Wenn man die Register des Timers direkt programmiert, wie wir das später tun werden, haben wir die volle Kontrolle über das Geschehen. Anders – aber viel einfacher – sieht es aus, wenn wir die Bequemlichkeiten des Arduino nutzen. Im Grunde benötigen wir nur das Kommando `analogWrite()`, geben als Parameter

den Ausgabepin (immer im Rahmen des Möglichen bei diesem Controller) und das Tastverhältnis zwischen 0 und 255 an. Das war es auch schon. Der kurze Arduino-Sketch sieht dann z. B. so aus:

```
// Arduino UNO
// PMW-Ausgabe

int pin = 9; // Ausgabe auf OC1A

void setup()
{
    analogWrite(pin, 127); // Tastverhältnis 50% (255/2 = 127)
}

void loop() {}
```

Hier machen die Arduino-Bibliotheken die meiste Arbeit. Das Tastverhältnis lässt sich jedoch nicht in max. 65536 Schritten, sondern nur in 255 Stufen einstellen. Meist reicht das, aber es geht besser. Auch die Frequenz ist mit ca. 500 Hz fix vorgegeben. Das resultierende Rechtecksignal zeigt



Tiefpass für den Arduino UNO
PWM-Rechteck -> Gleichspannung

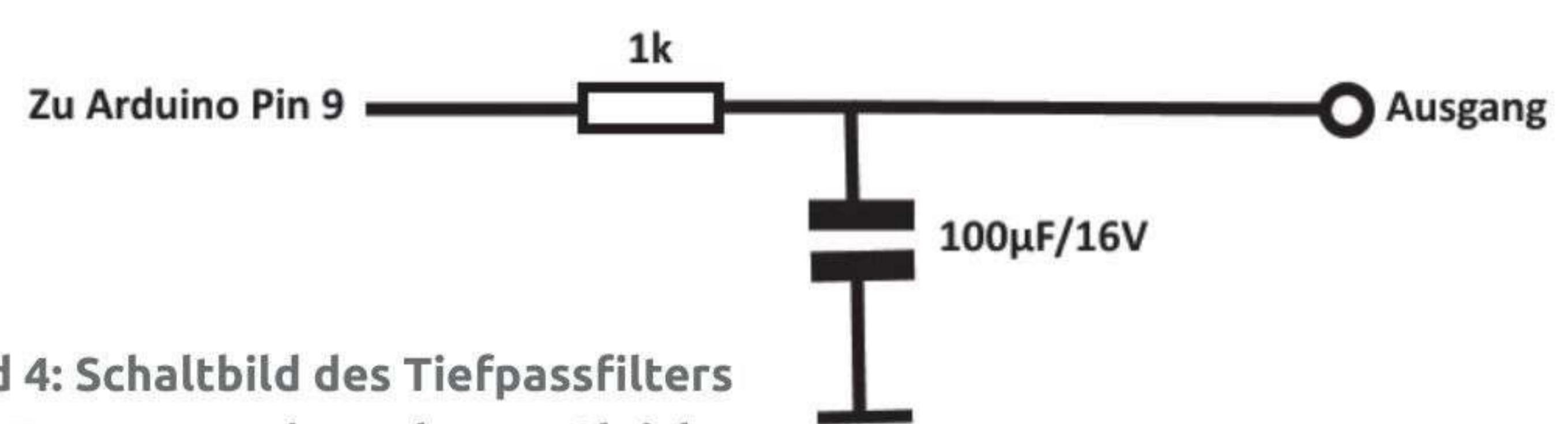


Bild 4: Schaltbild des Tiefpassfilters zur Erzeugung einer glatten Gleichspannung zur Steuerung von Geräten mit analogem Eingang.

Bild 2. Wie es sich bei anderen Werten als 127 (für ein 50-prozentiges Tastverhältnis) ändert, zeigt Bild 3. Mit dem Arduino kommen wir zwar schnell ans Ziel, geben allerdings einige Kontrolle auf. Wenn nun aus diesem Rechtecksignal eine analoge Spannung werden soll, werden ein Widerstand von 1k und ein Elko von 100μF benötigt und wie folgt an Pin 9 des Arduino angeschlossen (Bild 4).

Nun messen wir eine Gleichspannung von 2,52 Volt mit dem Oszilloskop (oder einem Multimeter).

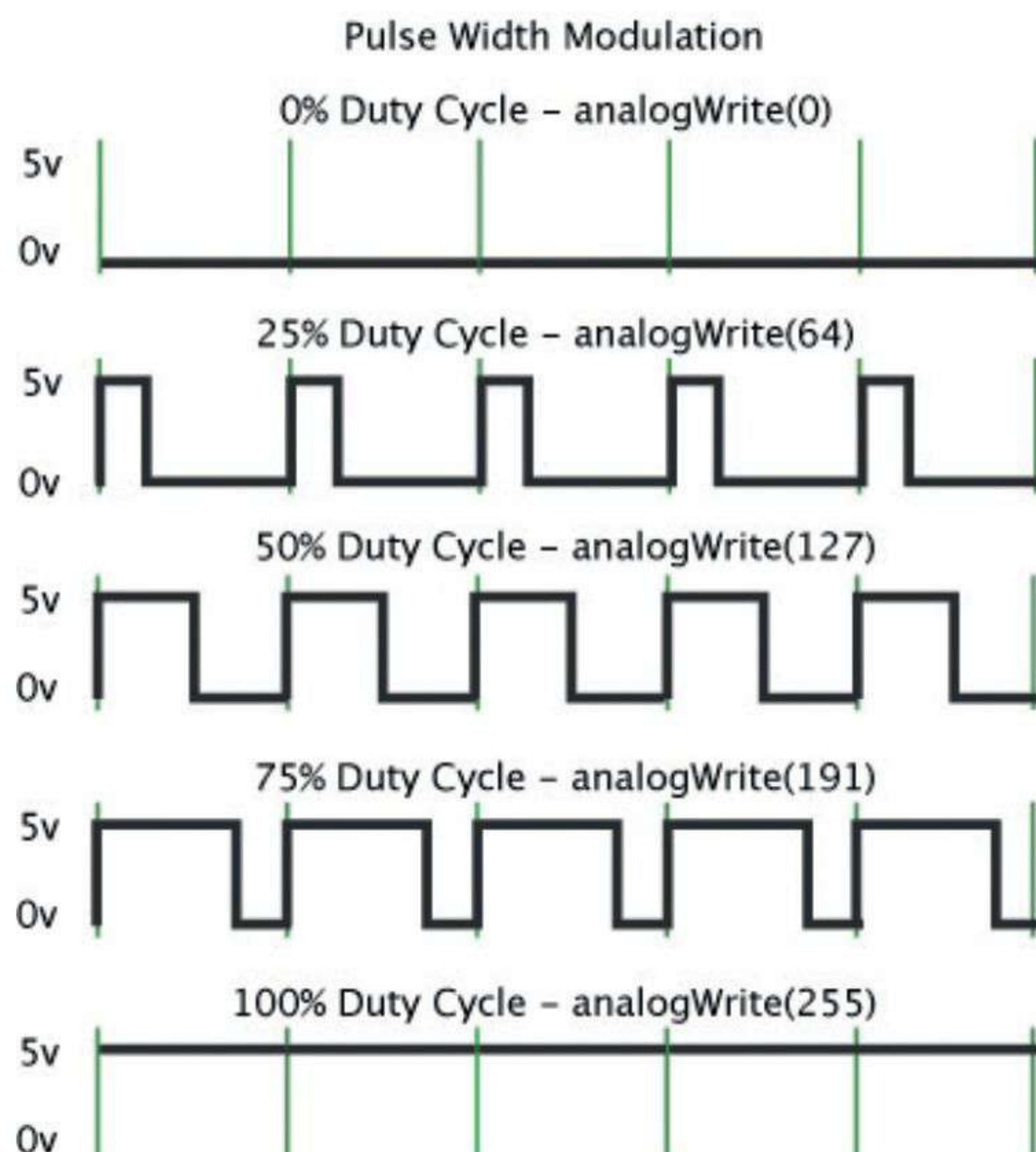


Bild 2: Der Arduino UNO erzeugt ein PWM-Signal, Tastverhältnis 50%, 1ms High-Pegel und 1ms Low-Pegel.

Bild 3: Einstellung des Tastverhältnisses beim Arduino-Kommando `analogWrite()`. Quelle: arduino.cc.

Die Pulswidtemodulation

Nehmen wir an, dass unsere Zählervariable bei Null anfängt. Mit jedem eingehenden Takt wird die Variable inkrementiert (um eins erhöht) und hat nach einer Sekunde einen Wert von etwa 16 Millionen erreicht (bei 16 MHz Takt des Arduino UNO). Nun, das würde er, wir verwenden aber nur einen 16-Bit-Timer, der von 0-65535 zählen kann. Bei 8 MHz Takt erreicht der Zähler seinen höchsten Stand schon nach ungefähr 8 ms, bei 16 MHz schon nach 4 ms. Gelangt der Zähler an seine obere Grenze, setzen wir den Zähler auf Null zurück und zählen einfach weiter. In einer Sekunde und bei 4 ms Zykluszeit der Zähler etwa 244 Mal (16 MHz / 65536 = 244,14 Hz) seinen Zyklus durchlaufen. Um den Zähler dazu zu bringen, etwas Interessantes zu tun, verwenden wir eine zusätzliche Variable, die in einem Vergleichsregister abgelegt ist und die dem Timer mitteilt, wann er seinen digitalen Ausgabepin (z. B. OC1A) ändern soll: Wenn der Zähler bei Null anfängt zu zählen, gibt sein physikalischer Ausgangspin einen logischen Low-Pegel (Null Volt) aus. Sobald die Zählervariable gleich dem Wert im Vergleichsregister ist, wird der Ausgangspin auf High geschaltet (z.B. 5 V) und sobald der Zähler an seiner oberen Grenze von 65635 auf Null zurückgesetzt wird, geht der Ausgangspin wieder auf Low. Um die Sache noch interessanter zu gestalten, können wir diese Obergrenze auch auf eine Zahl kleiner als 65535 verändern, wodurch sich die Gesamthäufigkeit, mit der sich ein Zyklus wiederholt, ändert.

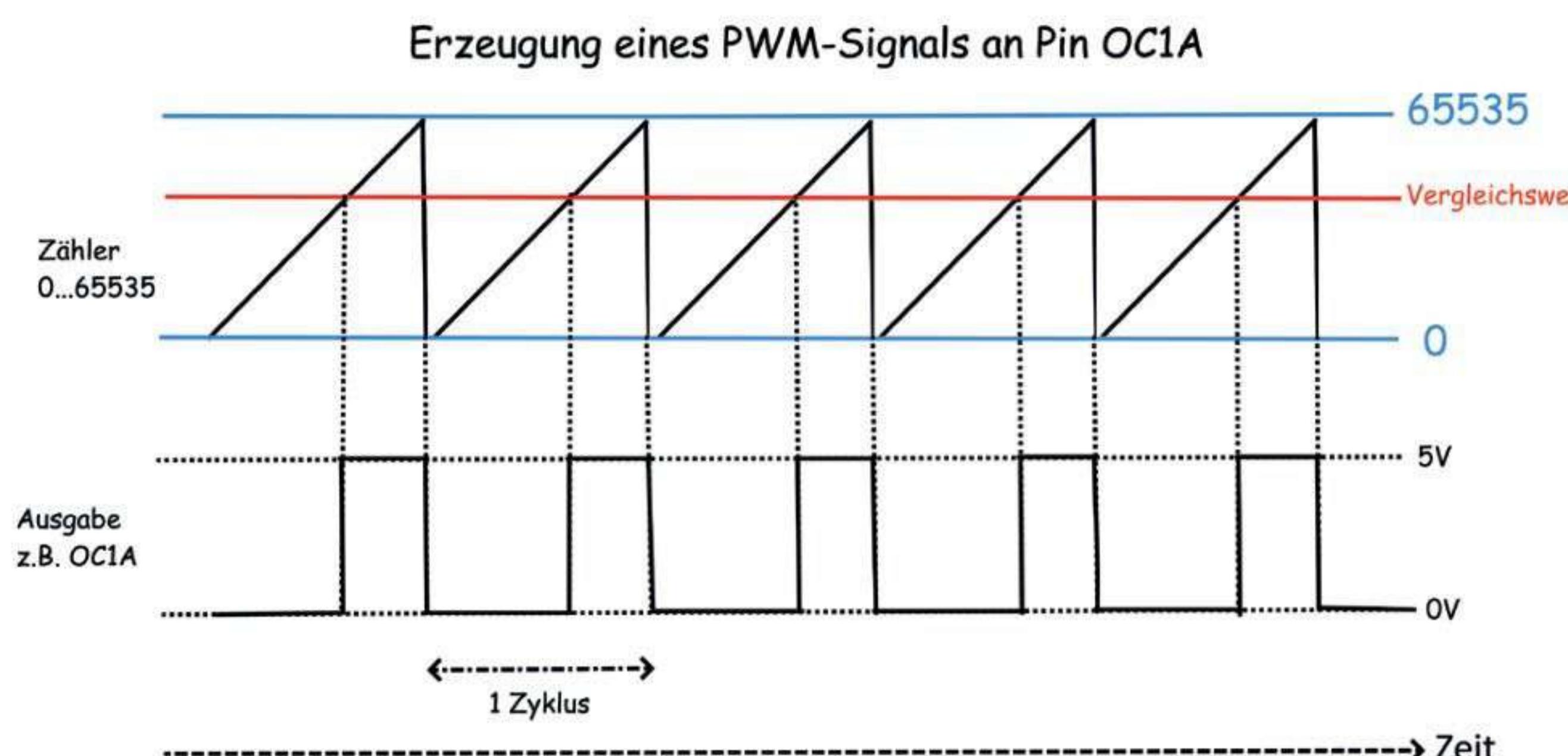


Bild 5: Erzeugung eines PWM-Signals: Erreicht der Zähler den Vergleichswert, schaltet der Ausgabepin des Prozessors auf High-Pegel und bei Erreichen von 65535 auf Low-Pegel.

Einige konkrete Beispiele

Nehmen wir an, dass wir mit dem internen 8-MHz-Takt zählen, die obere Zählgrenze 65535 ist und das Vergleichsregister auf 32767 gesetzt ist. Dann wird während jedes Zyklus bei Zählerwerten von Null bis 32766 der Ausgangspin Low-Pegel aufweisen, bis der Zähler auf 32767 steht, nun schaltet der Pegel auf High, bis er auf 65535 steht und wieder auf Null zurückgesetzt wird. Dabei wird auch der Ausgangspin auf Null gesetzt (Bild 5). Die Ausgangswellenform ist dann eine Rechteckwelle mit 50 Prozent Tastverhältnis (auf der ersten Hälfte der Zeit) bei einer Frequenz nahe 122 Hz.

Als nächstes könnten wir den Wert des Vergleichsregisters auf 45000 ändern. Dies führt dazu, dass der Ausgang etwa 30 Prozent der Zeit eingeschaltet ist bei 122 Hz. Auch hier geht der Zähler von 0 bis 65535, er zählt mit 8 MHz und setzt sich zurück, wenn er die obere Grenze erreicht. Wenn der Zähler den Wert im Vergleichsregister (45000) passiert, wird der Ausgang High. Beim Zurücksetzen des Zählers auf Null geht der Ausgang wieder auf Low. Die resultierende Ausgangswellenform hat ein Tastverhältnis von etwa 30 Prozent (30 % High und 70% Low).

Wenn wir nun die obere Zählgrenze von 65535 auf 32767 ändern und den Vergleichswert auf 16383, durchläuft der Zähler einen kompletten Zyklus in nur 4 ms, die Taktfrequenz beträgt also ungefähr 244 Hz. Da das Vergleichsregister wieder auf die Hälfte der oberen Zählgrenze gesetzt wird, beträgt das

Tastverhältnis wiederum 50 Prozent. Der durch ein nachfolgendes Tiefpassfilter gebildete Mittelwert des Ausgangssignals kann von Null bis 100 Prozent des logischen Ausgangspegels variieren (z. B. 0 bis 5 V), und zwar durch Variation der zeitlichen Breite der regelmäßigen Ausgangsimpulse, daher der Name Pulsweitenmodulation.

Diese Methode der Digital-Analog-Wandlung wird nie besonders präzise sein, so dass die Verwendung unserer vollen 16 Bit Auflösung zur Angabe des Tastverhältnisses etwas übertrieben ist. Stattdessen werden wir zehn Bits verwenden, was uns 1024 mögliche unterschiedliche Ausgangspegel gibt. Immerhin das Vierfache des Arduino! Indem wir dieses Opfer bei der Auflösung bringen, können wir etwas Geschwindigkeit gewinnen. Wir setzen die obere Zählgrenze auf 1023, was bedeutet, dass wir einen vollen Zählzyklus in nur 128 Mikrosekunden abschließen können, was uns eine respektable Ausgangsfrequenz von 7,8 kHz beschert.

Ran an die Register!

Da wir nun wissen, was der Timer tun soll, müssen wir nur noch dafür sorgen, dass der Chip diese Dinge auch tatsächlich ausführt. Wir müssen zuerst die Taktquelle und den Teiler einstellen, danach die obere Zählgrenze festlegen, als Drittess müssen wir einen Anfangswert für das Vergleichsregister festlegen (und vielleicht dazu die obere Zählgrenze) und Viertens die Timer-Steuerregister so einstellen, dass er in dem von

uns beschriebenen Modus arbeitet.

Beginnen wir mit dem Takt. Es gibt zwei, potentiell unterschiedliche, Taktsignale, um die wir uns kümmern müssen. Das erste ist die Systemtaktfrequenz des Mikrocontrollers und das zweite ist das Eingangssignal für die Zähler-/Timerschaltung. Wir werden die Systemtaktfrequenz als Eingangssignal für den Timer verwenden, so dass diese beiden Frequenzen gleich sind. Da wir einen Arduino UNO als Hardware nutzen wollen, ist der Systemtakt mit 16 MHz vorgegeben. Es ist möglich, mit einer niedrigeren Rate zu arbeiten, indem man einen Clock-Prescaling-Teiler verwendet, aber wir wollen diese Funktion ausschalten. Als nächstes müssen wir uns um die Wahl der Taktquelle für den Timer kümmern. Die Dokumentation des 16-Bit-Timers befindet sich in Kapitel 14 des Datenblattes, und die Taktquelle wird durch das „Timer/Counter1 Control Register B“, oder TCCR1B, gesteuert. Hier sehen Sie, wie TCCR1B aussieht:

TCCR1B – Timer/Counter1 Control Register B

Bit (0x81)	7	6	5	4	3	2	1	0	TCCR1B
Read/Write	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	
Initial Value	0	0	0	0	0	0	0	0	

Bild 6: Definition des Timer-Registers TCCR1B zur Einstellung u.a. für den Takt. Quelle: Datenblatt.

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1X at	TOV1 Flag Set on
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Bild 7: Einstellungen zum Wave generator Mode (WGM). Quelle: Datenblatt.

Die untersten drei Bits heißen CS10, CS11 und CS12, für Clock Select [counter#] 1, Bits 0-2. Aus einer Tabelle im Datenblatt sehen wir, dass diese drei Bits auf 001 (dezimal 1) gesetzt werden sollten, um den Systemtakt (hier „I/O-Takt“ genannt) ohne vorgeschalteten Teiler zu verwenden.

Was ist sonst noch im Register TCCR1B? Die Bits 6 und 7 sind für die Verwendung des Timers zur Messung externer Signale bestimmt. Wir werden diese nicht verwenden, also können wir die Standardwerte von Null belassen. Die anderen beiden Bits in TCCR1B, die etwas bewirken (WGM12 und WGM13), sind

Steuerbits für den Timer-Modus. Sie sind zwei der vier Steuerbits für den Zähler/Timer *Waveform Generation Mode*, wobei die anderen beiden Bits in einem der anderen Steuerregister erscheinen.

Aus Bild 7 geht hervor, dass der Modus, den wir verwenden wollen, die schnelle PWM ist, bei der der Zähler unidirektional von 0 bis zu einem einstellbaren Maximalwert (top) inkrementiert und dann wieder bei Null beginnt, das ist der Modus Nummer 14.

Der Binärwert für diesen Modus ist 1110b (dezimal 14), also sollten WG13 und WG12 beide auf 1 gesetzt werden. Alles zusammengekommen wollen wir TCCR1B auf 00011001b, oder 25 dezimal, setzen. In C sieht der Code zum Setzen dieses Registers folgendermaßen aus:

TCCR1B = 25;

Die andere Hälfte der Einstellungen für den Wellenform-Erzeugungsmodus wird (überraschenderweise) im „Timer/Counter1 Control Register A“ oder TCCR1A gehalten:

TCCR1A – Timer/Counter1 Control Register A

Bit (0x80)	7	6	5	4	3	2	1	0	TCCR1A
Read/Write	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10	
Initial Value	0	0	0	0	0	0	0	0	

Bild 8: Einstellungen zur Wirkungsweise der Output Control Pins OC1A und OC1B. Quelle: Datenblatt.

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match, set OC1A/OC1B at BOTTOM (non-inverting mode)
1	1	Set OC1A/OC1B on Compare Match, clear OC1A/OC1B at BOTTOM (inverting mode)

Bild 9: OC1A und OC1B sollen High werden, sobald der Zähler den Vergleichswert erreicht und Low, wenn der Zählen nach einem Überlauf auf Null springt. Quelle: Datenblatt.

Alles zusammengenommen wollen wir TCCR1A auf 11110010b oder 242 dezimal setzen. In C lautet der Code zum Setzen dieses Registers wie folgt:

TCCR1A = 242;

An diesem Punkt haben wir die Systemtaktfrequenz und die Timer-Taktfrequenz eingestellt und wir haben die Ausgangs- und Vergleichsmodi entsprechend eingestellt. Da wir den Wellenform-Erzeugungsmodus vierzehn gewählt haben, ist der obere Wert, bei dem der Zähler zurückgesetzt wird, durch den Wert gegeben, der in ICR1, dem Input-Capture-Register, gespeichert ist.

Man muss beim Setzen von ICR1, wie auch bei den Vergleichsregistern, etwas vorsichtig sein, da es sich um echte 16-Bit-Register handelt und die High- und Low-Bytes in der richtigen Reihenfolge geschrieben werden müssen, wenn man dies per Assembler macht. Ihr C-Compiler (in der Arduino-IDE) wird das für Sie regeln. Wir schreiben also:

ICR1 = 1023;

Nun müssen die Vergleichsregister OCR1A und OCR1B auf Werte gesetzt werden, welche die Tastverhältnisse für die beiden Ausgänge festlegen. Beginnen wir damit, Ausgang A auf 50 Prozent und B auf 80 Prozent Tastverhältnis zu setzen. Beachten Sie, dass 80 Prozent Tastverhältnis bedeutet, dass der Ausgang nach 20 Prozent des Zyklus eingeschaltet wird:

OCR1A = 511;

OCR1B = 204;

Das letzte Detail

Wir sind fertig, oder? Nicht ganz. Wir haben den Timer so eingerichtet, dass er korrekt funktioniert, aber wir haben unserem Mikrocontroller noch nicht gesagt, dass er die Signale des Timers tatsächlich ausgeben soll. Wie wir bereits erwähnt haben, befinden sich OC1A und OC1B an den Pins 15 bzw. 16 des Prozessors (Arduino-Pins 9 und 10). Diese Pins sind nicht speziell für die Verwendung mit dem Timer vorgesehen, sondern sie werden gemeinsam mit dem Port B verwendet, und zwar mit den Pins PB1 und PB2. Ob ein einzelner Pin als Eingang oder Ausgang konfiguriert ist, wird durch das Port B Data Direction Register, DDRB, gesteuert, ein 8-Bit-Register, das für jeden der acht Pins in Port B eine 0 (Eingang) oder 1 (Ausgang) enthält. DDRB ist standardmäßig auf alle Nullen eingestellt, so dass alle Pins Eingänge sind. Um die Pins PB1 und PB2 als Ausgänge für den Timer zu verwenden, müssen wir sie als Ausgänge einstellen, wie in Abschnitt 12.3.2 des Datenblatts beschrieben. Dazu setzen wir die DDRB-Bits 1 und 2 (für PB1/OC1A und PB2/OC1B) und lassen die anderen auf Null, so dass wir 00000110b, oder 6 dezimal. Der C-Code dafür ist

DDRB = 6;

Eine etwas schönere Art, dies zu tun, ist die Verwendung des avr-libc-Makros `_BV(bit)`, das dem C-Befehl `(1<<bit)` entspricht, mit dem Unterschied, dass es zur Komplierzeit ausgeführt wird, so dass

keine Zeit mit der Berechnung der Bitverschiebung verschwendet wird, während der Mikrocontroller sein Programm ausführt. Wir können dann unseren C-Befehl schreiben als

DDRB = `_BV(PB1) | _BV(PB2);`

wobei PB1 und PB2 als 1 bzw. 2 in io.h definiert sind. Die Anweisung ist dann äquivalent zu `DDRB = (1 <<1) | (1 <<2) = 2 | 4 = 6`, hat aber den Vorteil, dass sie etwas einfacher zu ändern ist als eine harte und feste Zahl. Die Gesamtheit unseres Codes, um die beiden pulsweitenmodulierten Signale mit 50 Prozent und 80 Prozent Tastverhältnis zu erzeugen, sieht wie folgt aus, und er ist ziemlich kurz:

```
TCCR1A = 242;
TCCR1B = 25;
ICR1 = 1023;
OCR1A = 511;
OCR1B = 204;
DDRB = _BV(PB1) | _BV(PB2);
```

Nachdem ich den Arduino UNO mit diesem Code programmiert habe, habe ich mir den Ausgang auf meinem Oszilloskop angesehen.



Bild 10: Ausgabe der PWM über OC1A und OC1B. Oben die 50%-Tastung, darunter das Signal von OC1B mit einem Tastverhältnis von 80 Prozent.

Die hier gezeigten Signale (Bild 10) sind oben dasjenige mit 50 Prozent Tastverhältnis, darunter das mit 80% Tastung. Beide beschreiben ein sauberes Rechtecksignal. Mit einer Skala von 20 Mikrosekunden pro Teilung misst das Oszilloskop eine Frequenz von 15,62 kHz ergibt, ein Wert, der innerhalb eines Prozentes der nominalen 16 kHz liegt, die wir erwartet haben.

Bislang haben wir die Erzeugung und Ansteuerung von Pulsweitenmodulationssignalen demonstriert. Wenn Sie nur versuchen, eine LED mit dieser Technik zu dimmen, werden Sie nicht viel sehen, da das Auge das 16-kHz-Blinken nicht erkennen kann. Wir versehen beide Ausgänge nun mit einem RC-Filter. Beginnen wir mit einem einfachen Tiefpassfilter, bestehend durch einen 1k-Widerstand und einen Elko von 0,47µF, weil ich den Elko gerade zur Hand hatte).

An den Ausgängen der Tiefpassfilter messen wir mit einem Multimeter etwa 2,5 V für das Signal an OC1A und etwa 4 V für den anderen Ausgang. Das passt gut zu den gesetzten Tastverhältnissen von 50 und 80 Prozent von 5 V. Experimentieren Sie mit den Wertem des Widerstandes und des Kondensators und betrachten Sie die Veränderungen des Ausgangssignals auf einem Oszilloskop.

Damit ist die Simulation zweier analoger Ausgänge mittels PWM und Tiefpassfilter gelungen. Wenn man soweit ist und die Gleichspannung der beiden Ausgänge kontrollieren kann, bietet es sich an,

einen Schritt weiter zu gehen und ein simples Dreiecksignal quasi-analog zu erzeugen.

Zwei Mal Dreieck

Zum Schluss also ein simpler Funktionsgenerator, indem wir einfach den Ausgangs-Vergleichswert variieren, der im Timer eingestellt ist. Zur Veranschaulichung hier der Code

zur Erzeugung eines Paares von Dreieckssignalen:

```
// Arduino UNO / NANO / MICRO
// Erzeugung von zwei Dreiecksignalen, Ausgabe auf OC1A und OC1B
// nachgeschaltet: jeweils Tiefpassfilter: R = 1k und C = 0,47uF

unsigned int m;
unsigned char up_down;

void setup()
{
  TCCR1A = 242;
  TCCR1B = 25;
  ICR1 = 1023;
  DDRB = _BV(PB1) | _BV(PB2);
  m = 0;
  up_down = 0; //1 = hochzählen, 0 = herunterzählen
}

void loop()
{
  delayMicroseconds(10); // Frequenz des Dreiecks hier einstellbar
  if (up_down) // hochzählen?
  {
    if (m < 1023) //ja
    {
      m++;
    }
    else
    { // 1023 erreicht, nun herunterzählen
      m--;
      up_down = 0;
    }
  }
  else // beim herunterzählen
  {
    if (m > 0)
    {
      m--;
      // kleiner werden, solange > 0
    }
    else
    { // ist Null, nun wieder größer werden
      m++;
      up_down = 1;
    }
  }
  OCR1A = m; // Berechnete Registerwerte zuweisen
  OCR1B = (1023 - m); // invertierter Wert
}
```

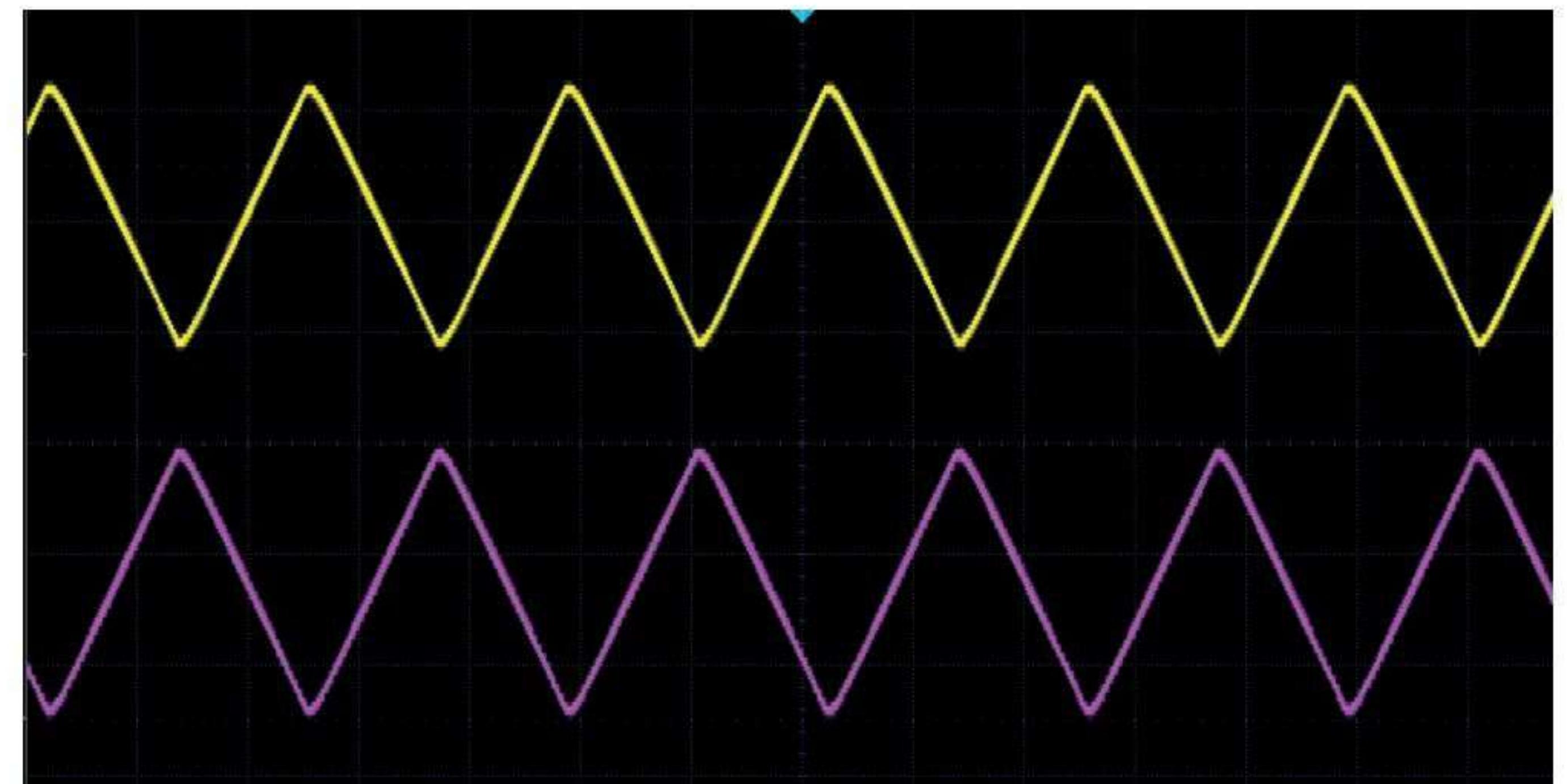


Bild 11: Zwei Dreiecksignale als kleine Demonstration eines Funktionsgenerators.

Die Werte im Register OCR1A/OCR1B werden verändert, sodass sich eine größere bzw. kleinere Ausgangsspannung ergibt. Wird der obere Wert erreicht, wird umgeschaltet und der Wert verkleinert. Ist die untere Grenze (null) erreicht, wird die Spannung wieder erhöht. Auf dem Oszilloskop sieht es nun aus wie in Bild 11.

Das war's: Ein einfacher und langsamer, aber nützlicher Funktionsgenerator. Stellen Sie sich vor, Sie erweitern die Hardware um zwei Potentiometer, die Sie an die analogen Eingänge des Arduino UNO anschließen, um Amplitude und Frequenz zu steuern. Die Erzeugung von Sinus-, Sägezahn- und Rechtecksignalen sind dem Leser als Programmierübungen überlassen. Einen leistungsstärkeren Funktions-

generator mit einem ESP32 finden Sie an anderer Stelle in diesem Heft. Die Arduino-Programme zu diesem Beitrag sind auf der virtuellen DVD im Ordner \Artikel\Atmel-DAC gespeichert.

Wer das Thema vertiefen möchte, findet auf Youtube.com eine Fülle von Videos. Das von User „Afug-Info“ mit dem Titel „Was ist PWM? Pulsweitenmodulation zu Sinus“ ist mir aufgefallen [1]. Es zeigt, wie PWM funktioniert und zu einer Sinuswelle wird.

Literatur/Verweise:

- [1] Video zu PWM: <https://www.youtube.com/watch?v=98si4uLcCpc>
- [2] Webseite zu Afug-Info: www.afug-info.de

Michael Wöste, DL1DMW

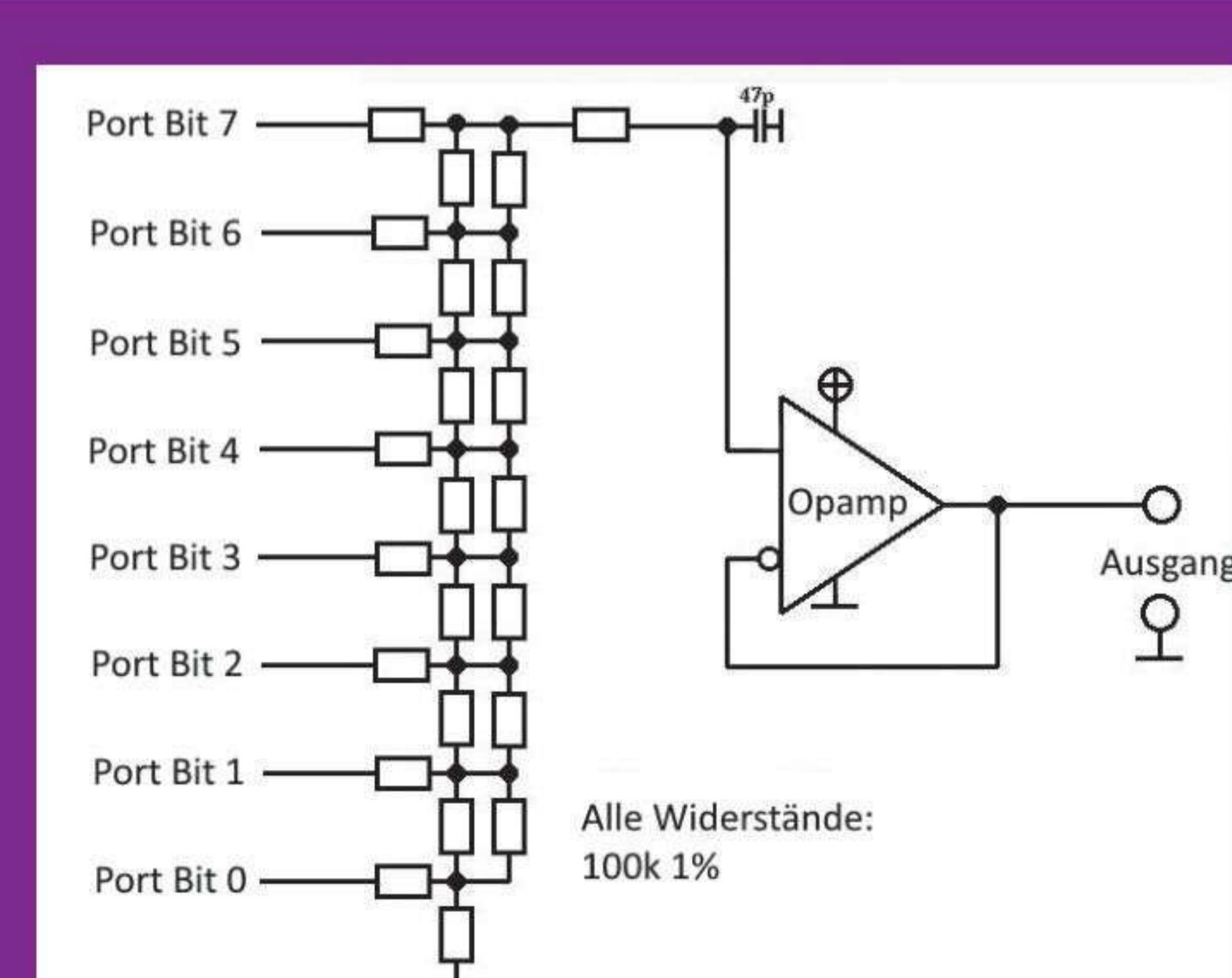


Bild 12: Prinzipschaltbild eines DAC, R/2R-Netzwerk.

mit einer größeren oder kleineren Zahl der Steuerbits realisieren. Je höher die Zahl der Bits, desto höher die Auflösung. Nehmen Sie beispielsweise 12 Bit, rechnet sich

seinen Teil zu der resultierenden Spannung bei. Bit 0, das niederwertige Bit, trägt 1/256 bei, Bit 1 trägt 1/128 bei, Bit 2 dann 1/64 usw. Sind alle Bits auf 1 gesetzt, liefert das Netzwerk die volle Ausgangsspannung. Die Werte der Widerstände sollten so genau sein, wie das gesamte Netzwerk exakt sein soll. Abweichungen sind besonders bei den höheren Bits relevant. Wenn also wie hier ein 8-Bit-DAC gewählt werden soll (256 Schritte), sind Widerstände mit 1% Genauigkeit erforderlich. Da diese Netzwerke wegen ihres hohen resultierenden Widerstands nicht viel Strom liefern kön-

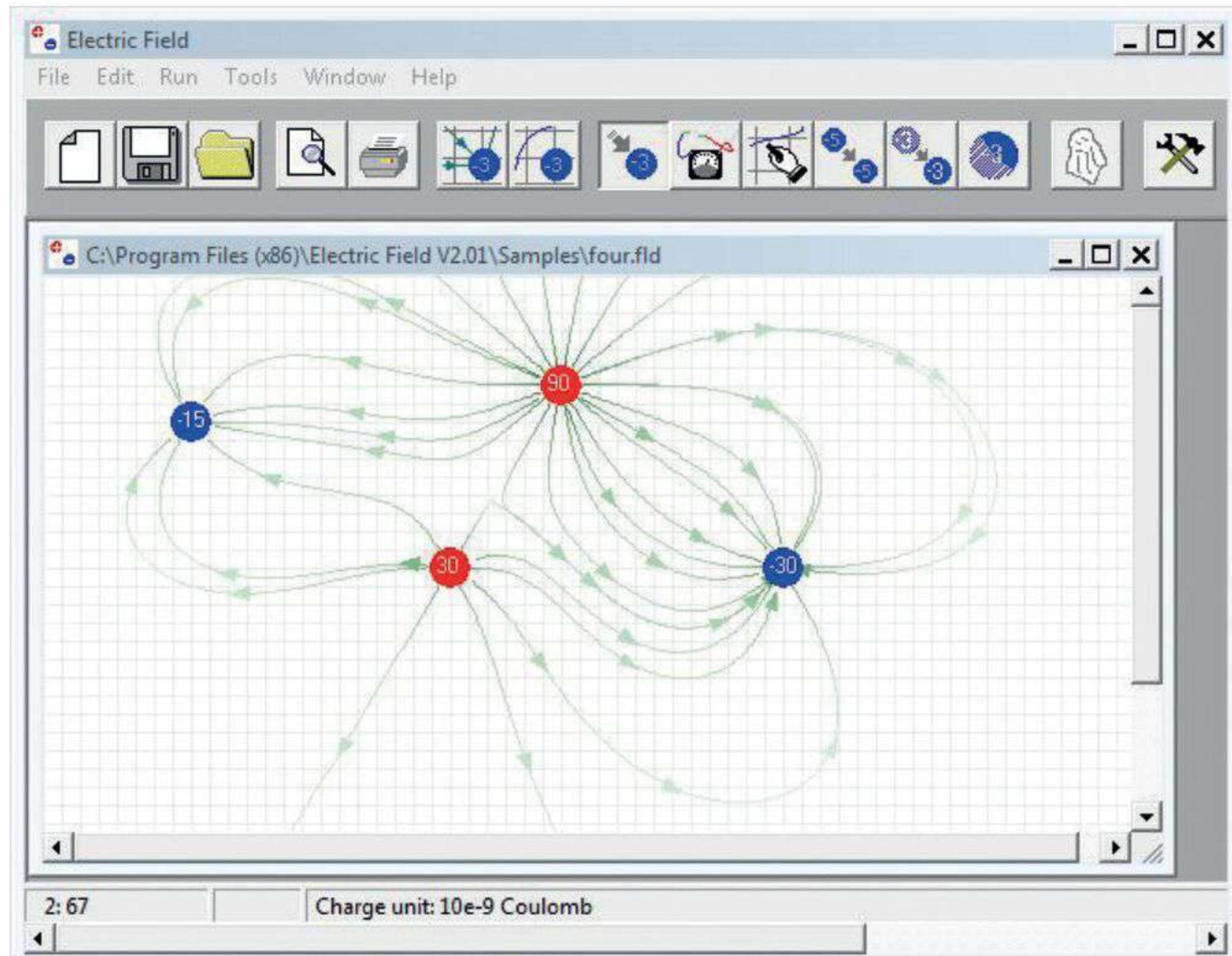
nen, entkoppelt ein Operationsverstärker das R2R-Netzwerk von der nachfolgenden Schaltung. Die Auflösung dieses Netzwerkes berechnet sich zu $5 \text{ Volt} / 256 \text{ Schritte} = 0,01953 \text{ Volt} = 19,5 \text{ mV}$. Das bedeutet: Legt man am Port des Controllers einen Wert von z. B. 100 an, misst die Spannung und danach 101, steigt die Spannung um die berechneten 19,5 mV. Das R2R-Netzwerk lässt sich nicht nur mit 8 Bit, sondern auch

die Auflösung bei fünf Volt zu $5 \text{ Volt} / 4096 \text{ Schritte} = 0,00122 \text{ Volt} = 1,22 \text{ mV}$. Aber: Wer hat schon zwölf freie Portpins nur für eine D-A-Wandlung frei? Noch ein Nachteil: Die Zahl der auf der Platine zu platzierenden Widerstände benötigt einiges an Platz. Und dann mag die PWM-Softwarelösung wieder als Alternative infrage kommen. Oder ein DAC in Form einer integrierten Schaltung.

DAC mit R2R-Netzwerk

Dieses Tutorial wäre nicht vollständig, würde es die Digital-zu-Analog-Wandlung mit einem R2R-Netzwerk verschweigen. Sind am Mikrocontroller beispielsweise acht freie Portpins frei, lässt sich diese Prinzipschaltung eines Widerstandsnetzwerkes nach Bild 12 aufbauen. Ein R2R-Netzwerk besteht aus Widerständen wie im Bild dargestellt. Die Bits, entweder bei 0 oder der Betriebsspannung, gehen über einen Widerstand mit doppeltem Wert als die vertikalen Teile des Netzwerks in die Ausgangsspannung ein. Jedes Bit trägt

seinen Teil zu der resultierenden Spannung bei. Bit 0, das niederwertige Bit, trägt 1/256 bei, Bit 1 trägt 1/128 bei, Bit 2 dann 1/64 usw. Sind alle Bits auf 1 gesetzt, liefert das Netzwerk die volle Ausgangsspannung. Die Werte der Widerstände sollten so genau sein, wie das gesamte Netzwerk exakt sein soll. Abweichungen sind besonders bei den höheren Bits relevant. Wenn also wie hier ein 8-Bit-DAC gewählt werden soll (256 Schritte), sind Widerstände mit 1% Genauigkeit erforderlich. Da diese Netzwerke wegen ihres hohen resultierenden Widerstands nicht viel Strom liefern kön-



Ladungen und Felder: FELDLINIEN simulieren und messen

Electric Field für Windows ist ein Simulator für elektrische Felder. Man positioniert ein oder mehrere Ladungen in Form von roten und blauen Punkten (Drähten in der Schnitt-Perspektive) auf eine Arbeitsfläche und startet die Simulation. Das Programm berechnet die Feldlinien eines geladenen elektrischen Drahtes. Sie können eine unbegrenzte Anzahl

von Ladungen in das Feld setzen: Fügt man mehrere Ladungen nebeneinander und parallel dazu untereinander, erhält man beispielsweise einen Kondensator. Electric Field berechnet auch Vektor und Betrag an bestimmten Punkten von Feldlinien (Bild 3).

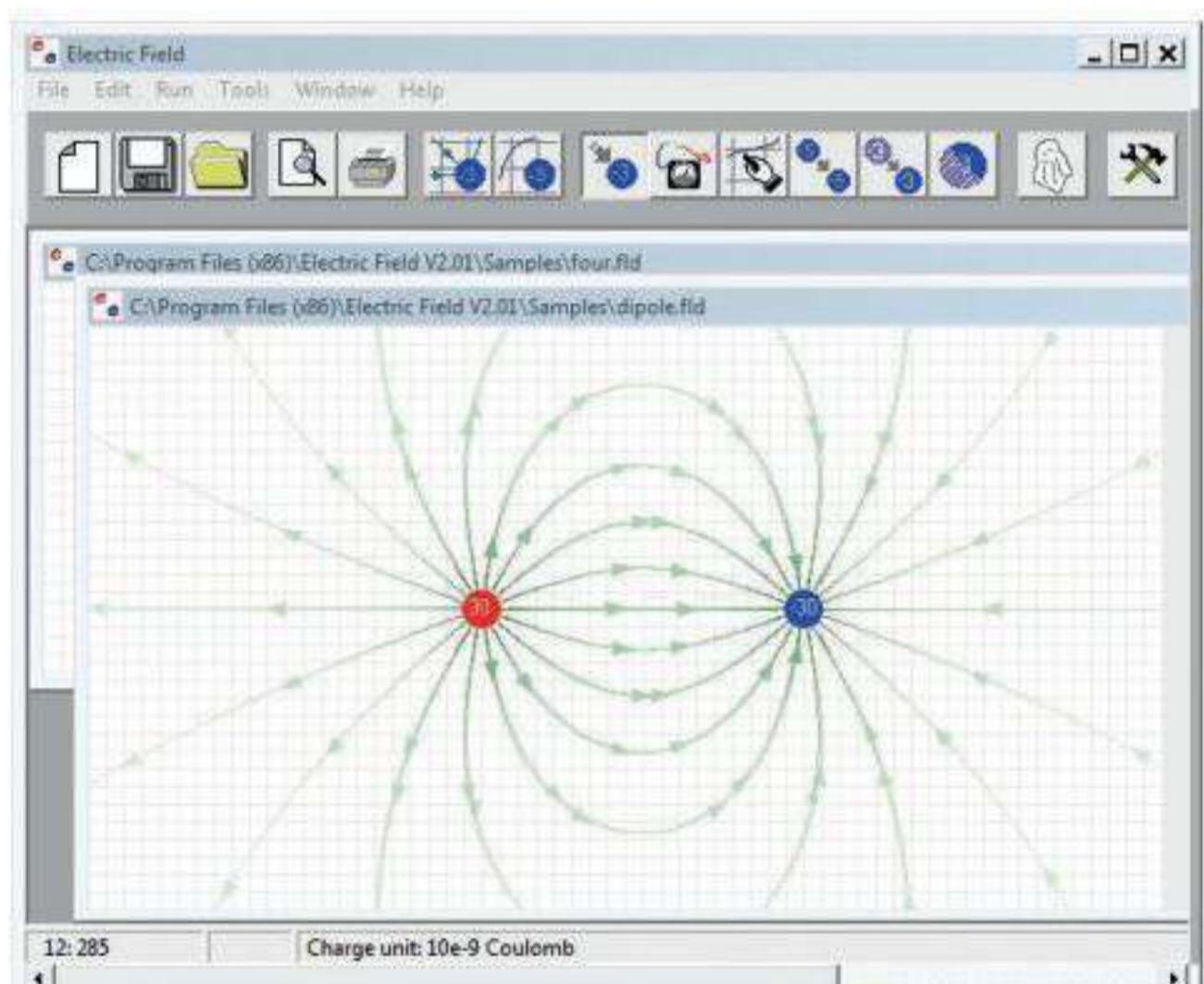


Bild 2: Feldlinien zwischen zwei Ladungsträgern.

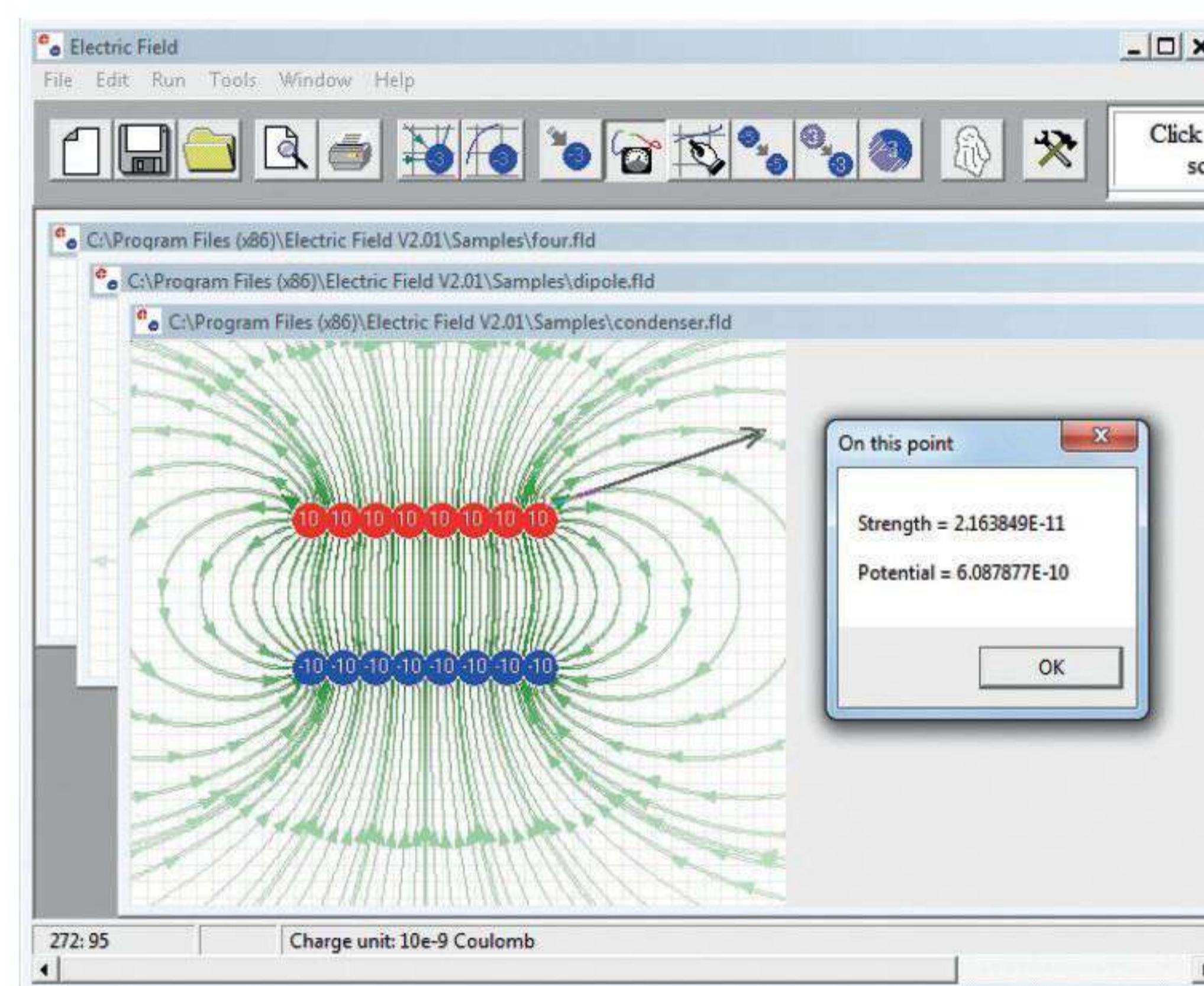
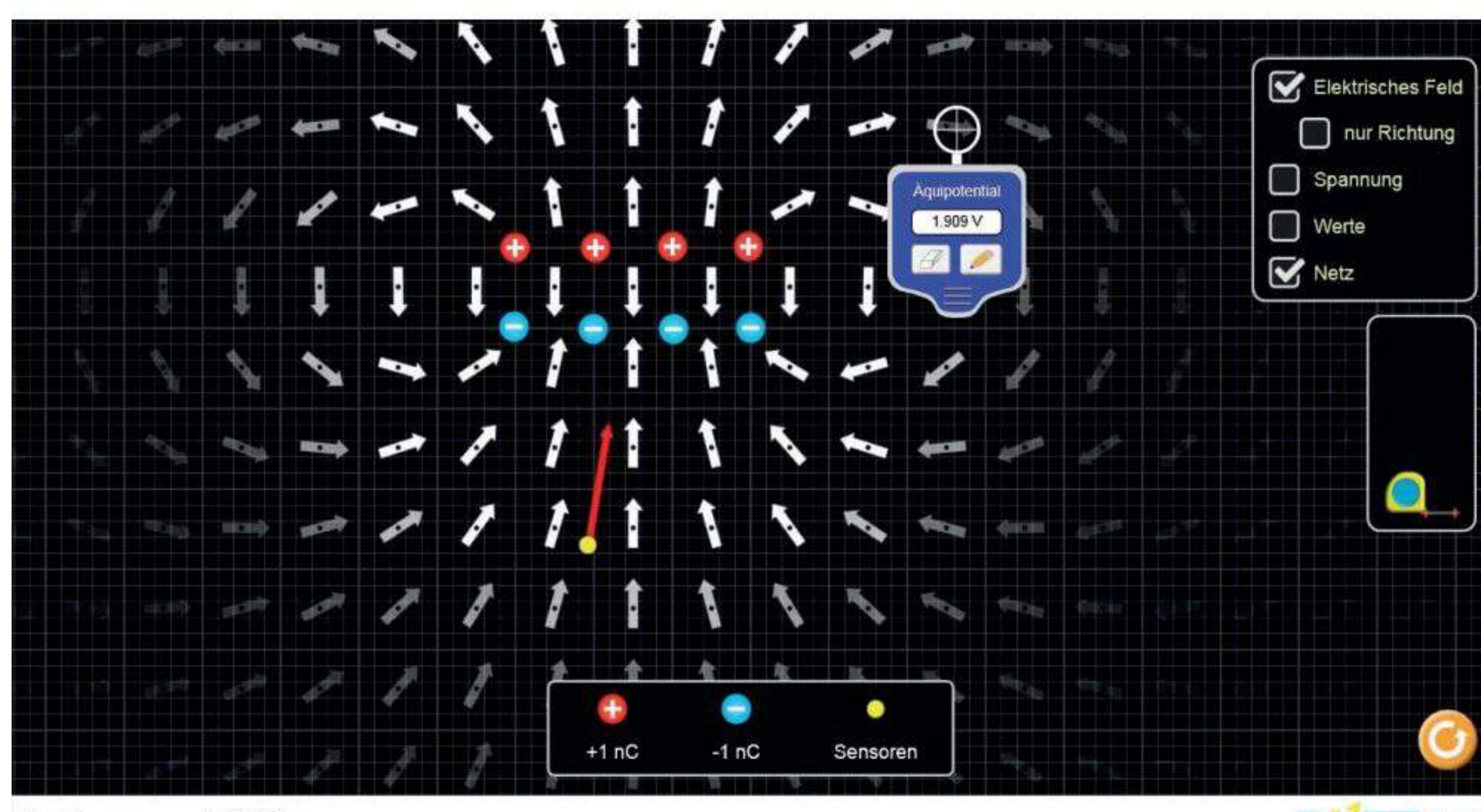


Bild 3: Ein Kondensator entsteht, wenn sich mehrere Ladungsträger gegenüberstehen. Mit dem Werkzeug Tester misst man Vektor und Betrag von Feldlinien aus.



Ladungen und Felder

Bild 4: Die Online-Simulation läuft im Browser, wenn man die Webseite [2] aufruft.

Bild 1: Simulation mehrerer positiver (roter) und negativer (blauer) Ladungen und deren Wirkung zueinander.

Dazu aktiviert man den *Tester* und fährt mit der Maus Feldlinien ab. An der Mausposition erscheint ein Pfeil, den Vektor und Betrag der Ladung symbolisiert. Echte Werte liefert das Programm nach einem Mausklick auf eine Feldlinie. Das Programm ist Freeware und darf kostenlos verteilt werden. Anwender, die viel mit der Software arbeiten, können auf der Webseite des Autors einen kleinen Betrag spenden. Selbstverständlich finden Sie das Programm auch auf der virtuellen DVD im Ordner *Simulation*.

Auf der Webseite [2] gibt es die Feldliniensimulation auch online im Browser. Auch hier positioniert man positive und negative Ladungen auf die Simulationsfläche, platziert Sensoren und ein Messgerät. Am Sensor, den man frei im Feld setzen kann, ist direkt der Vektor sichtbar, Werte dazu lassen sich einblenden.

Das Programm *Electric Field* ist schon etwas älter, sodass sich die Hilfedatei unter Windows 7 oder 10 nicht mehr laden lässt. Dennoch installiert sich die Anwendung unter den aktuellen Windows-Versionen einwandfrei. Es kann sinnvoll sein, Einsteigern einen ersten Eindruck von der Wirkung des Magnetismus eines Leiters zu vermitteln. Das gibt es auch als *Online-Simulation*, sodass man auf dem PC nichts installieren muss.

Von der Webseite [3] gelangt man zu weiteren Simulationen (Menü *Simulationen*, dann *Browser Sims* wählen) aus den Bereichen Physik, Biologie, Mathematik und Geowissenschaften. Eine Simulation eines simplen Gleichstromkreises, den sich der Anwender intuitiv aus verschiedenen Bauelementen zusammenklicken kann, zeigt Bild 5.

mw

Literatur/Verweise:

- [1] Electric Field: <http://www.physics-software.com/software.html>
- [2] Online-Tool: https://phet.colorado.edu/sims/html/charges-and-fields/latest/charges-and-fields_de.html
- [3] PHET-Homepage: <https://phet.colorado.edu/de/>

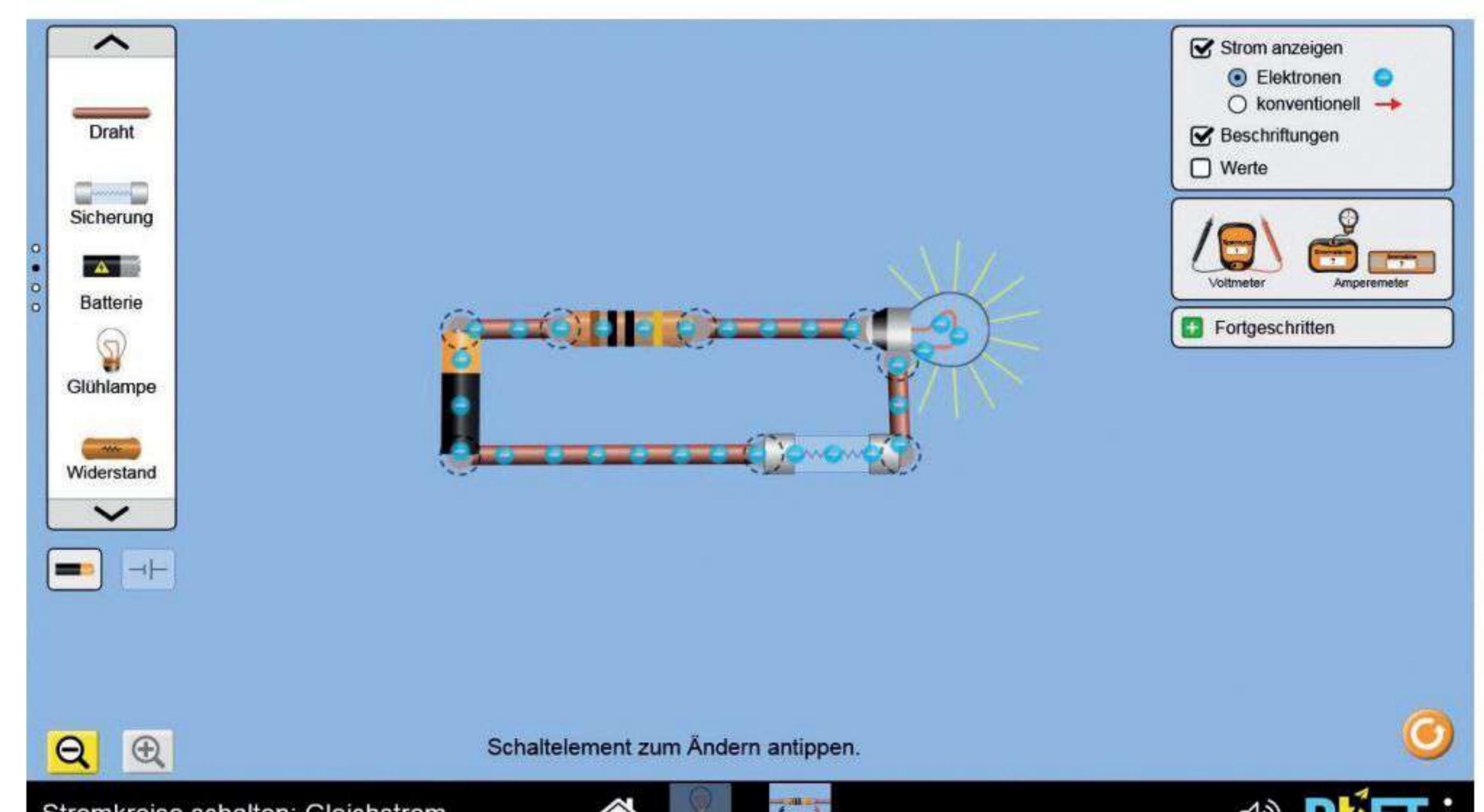


Bild 5: Es gibt viele Online-Simulationen, hier als Beispiel die eines Stromkreises.

Programmidee umgesetzt: Kleines Verzeichnis der Elektronik-Rechner

Die Idee zu dieser kleinen Freeware-Software ist diese: Berechnungsprogramme der Elektronik werden immer öfter als HTML-Software für den Browser realisiert und im WWW zur Verfügung gestellt, anstatt dass man sie auf den lokalen PC installiert. Bei der Vielzahl der Online-Rechner für die Elektronik – die in den Weiten des Internet verteilt sind – fällt die Übersicht schwer. Dieses Programm soll Abhilfe schaffen und sie an einem Ort bündeln.

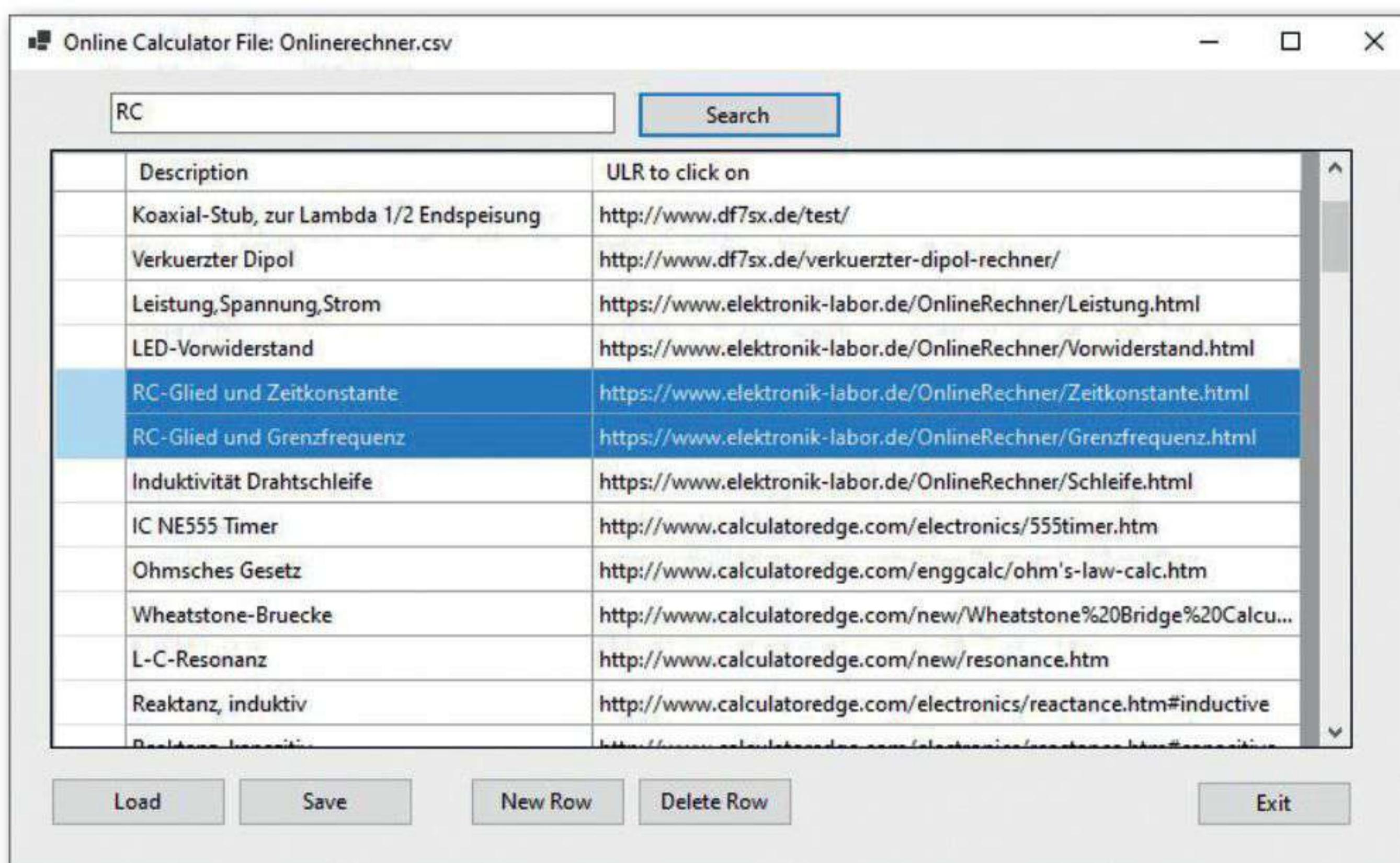


Bild 1: Kleines Verzeichnis der Online-Elektronikrechner.

Es handelt sich um ein kleines Programm für Windows 64-Bit, MacOS und Linux sowie für Windows 32-Bit in einer separaten Variante, das anhand einer Tabelle Internet-Adressen zahlreicher Online-Rechner verwaltet. In der linken Spalte der Tabelle findet man eine – möglichst aussagekräftige – Beschreibung des Rechners, links daneben die Adresse des Rechners im Internet (die URL). Klickt man auf die Zelle einer URL, startet der Webbrower und der betreffende Rechner – also die Webseite – erscheint auf dem PC-Monitor. Damit

ist die Entstehungsgeschichte des Programms auch schon hinreichend beschrieben.

Der Anwender kann und soll diese Liste individuell erweitern und taube – nicht mehr gültige – Links aus der Liste entfernen können. Zum Hinzufügen weiterer Online-Rechner dient der Button *New Row* (neue Zeile) und zum Entfernen einer Zeile aus der Tabelle die Schaltfläche *Delete Row* (lösche Zeile). Bevor man eine oder mehrere Zeile(n) löschen kann, muss man diese zuvor mit der Maus im Zeilenkopf vor der Beschreibung markieren. Ist die Zeile komplett markiert, betätigt man die Schaltfläche *Delete Row*.

Klickt man auf *New Row*, erscheint ein neues Fenster (Bild 2), das zwei

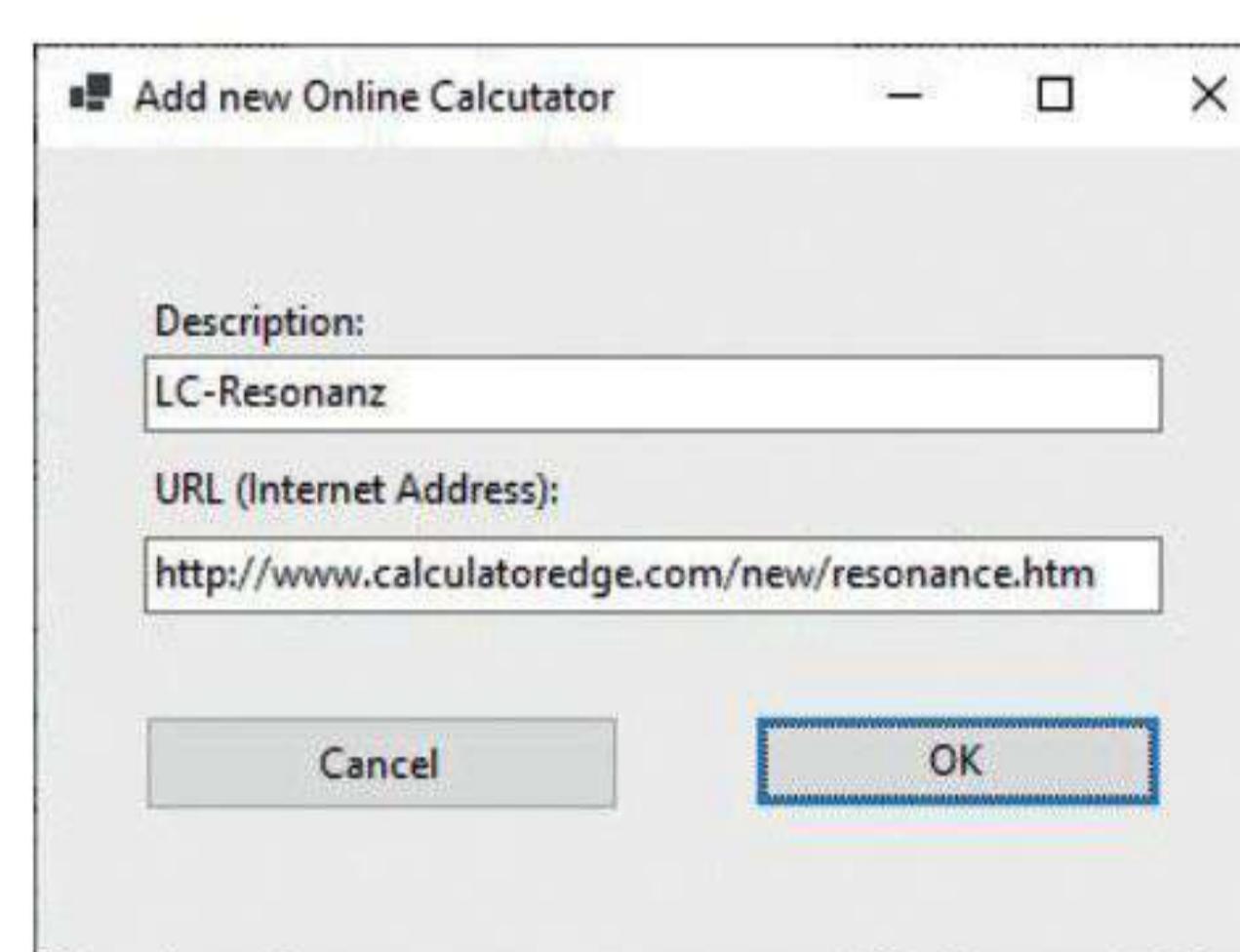


Bild 2: Eingabemaske für neue Tabelleneinträge.

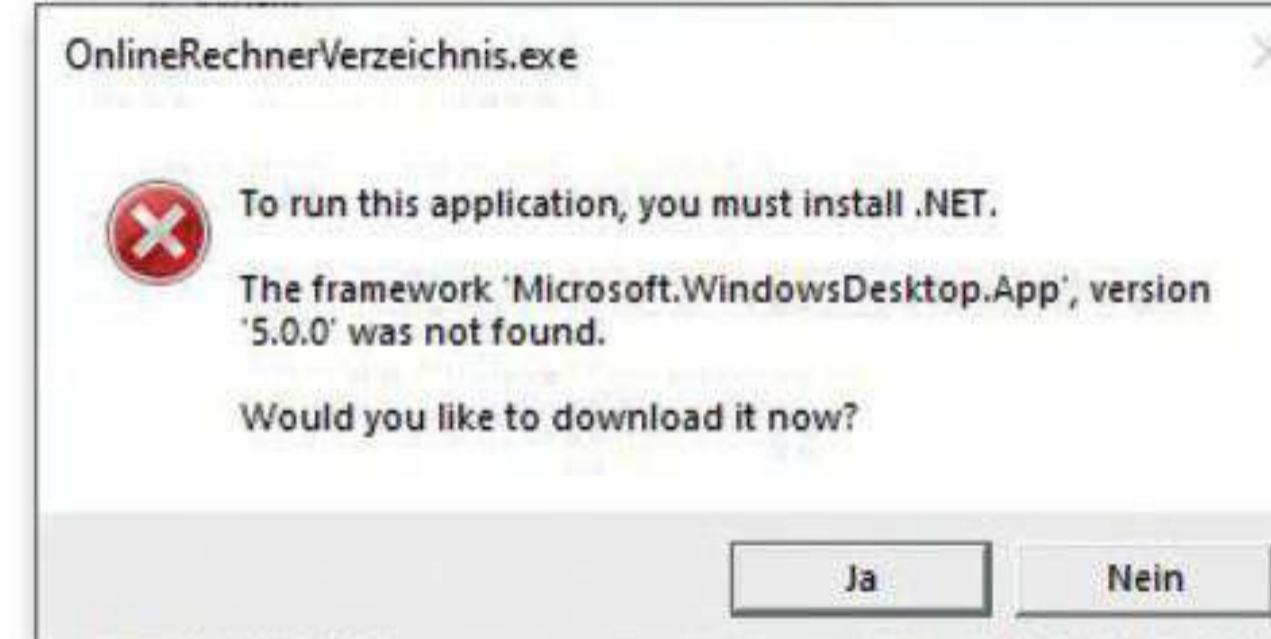


Bild 3: Beim Start der 64-Bit-Version bemerkt Windows, dass die .NET 5 fehlt und installiert werden muss.

Eingabefelder für die Beschreibung (oben) und die Internet-Adresse (unten) bietet. Die URL sollte man aus dem Browser über die Windows-Zwischenablage in das Feld einkopieren. In das Feld Beschreibung (*description*) setzt man einen „sprechenden“ Text, der den Online-Rechner gut beschreibt. Eine andere Möglichkeit ist das Eintragen des neuen Eintrages direkt in die letzte Zeile der Tabelle.

Schreibfehler bei der Beschreibung lassen sich – auch in der Tabelle – durch einen Mausklick auf die jeweilige Zelle und der anschließenden Betätigung der Return-Taste (auch Enter-Taste genannt) korrigieren. Nachdem man eine oder mehrere neue Einträge in die Liste gesetzt hat, sollte man sie unter der Bezeichnung *Onlinerechner.csv* mittels des Buttons

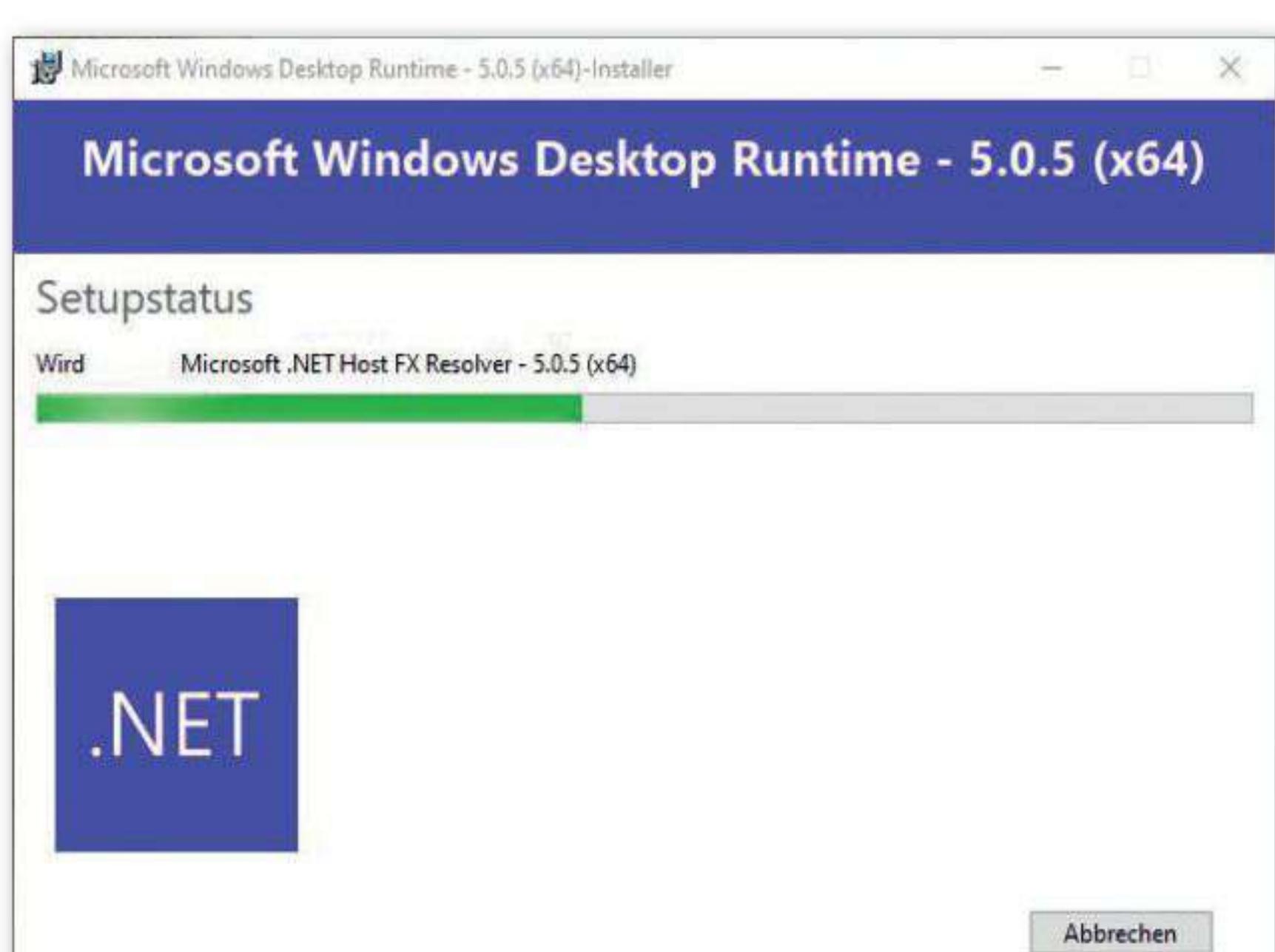


Bild 4: Download der 64-Bit-Version von .NET 5 bei Microsoft.

Save abspeichern. Diese Liste wird, sofern im selben Verzeichnis wie das Programm vorhanden, nach dem Start der Software selbsttätig geladen. Über *Load* kann man andere CSV-Dateien laden und so mehrere Link-Dateien verwalten. Neben der Liste der Online-Rechner könnten das beispielsweise Lieblings-Websiten sein, deren Adresse man zur Hand haben möchte. So kann dieses kleine Programm evtl. nützlich für allerlei Zwecke sein. Die Grundidee war jedoch, eine aktuelle Liste vieler Online-Rechner an einem Ort zur Verfügung zu haben.

Natürlich könnte man alle Links auch mit dem Browser verwalten. Sammeln sich sehr viele Links an, kann das unübersichtlich werden. Dabei hilft hier die Suchfunktion (*search*). Gibt man einen Suchbegriff – oder nur einige wenige Buchstaben – in das Eingabefeld ein und betätigt die Schaltfläche *Search*, markiert das Programm alle Zeilen der Tabelle blau, wenn der Suchbegriff mit der Beschreibung übereinstimmt. Das können durchaus mehrere Zeilen sein. Manche Treffer sieht man erst, wenn man die komplette Tabelle mit der Bildlaufleiste am Rand der Tabelle durchschaut. Mit einem Klick auf die URL gelangt man schließlich sehr schnell zum gewünschten Online-Rechner. Die Geschwindigkeit ist sicher ein Vorteil gegenüber einer Linkliste im Browser.

Unter Windows 64-Bit und andere Plattformen benötigt das Programm eine installierte Runtime-Version von Microsoft .NET 5, die man bei Microsoft lädt, sofern sie noch nicht auf dem PC installiert ist. Damit soll es auch unter MacOS oder Linux lauffähig sein, das wurde mangels Gelegenheit jedoch nicht getestet.

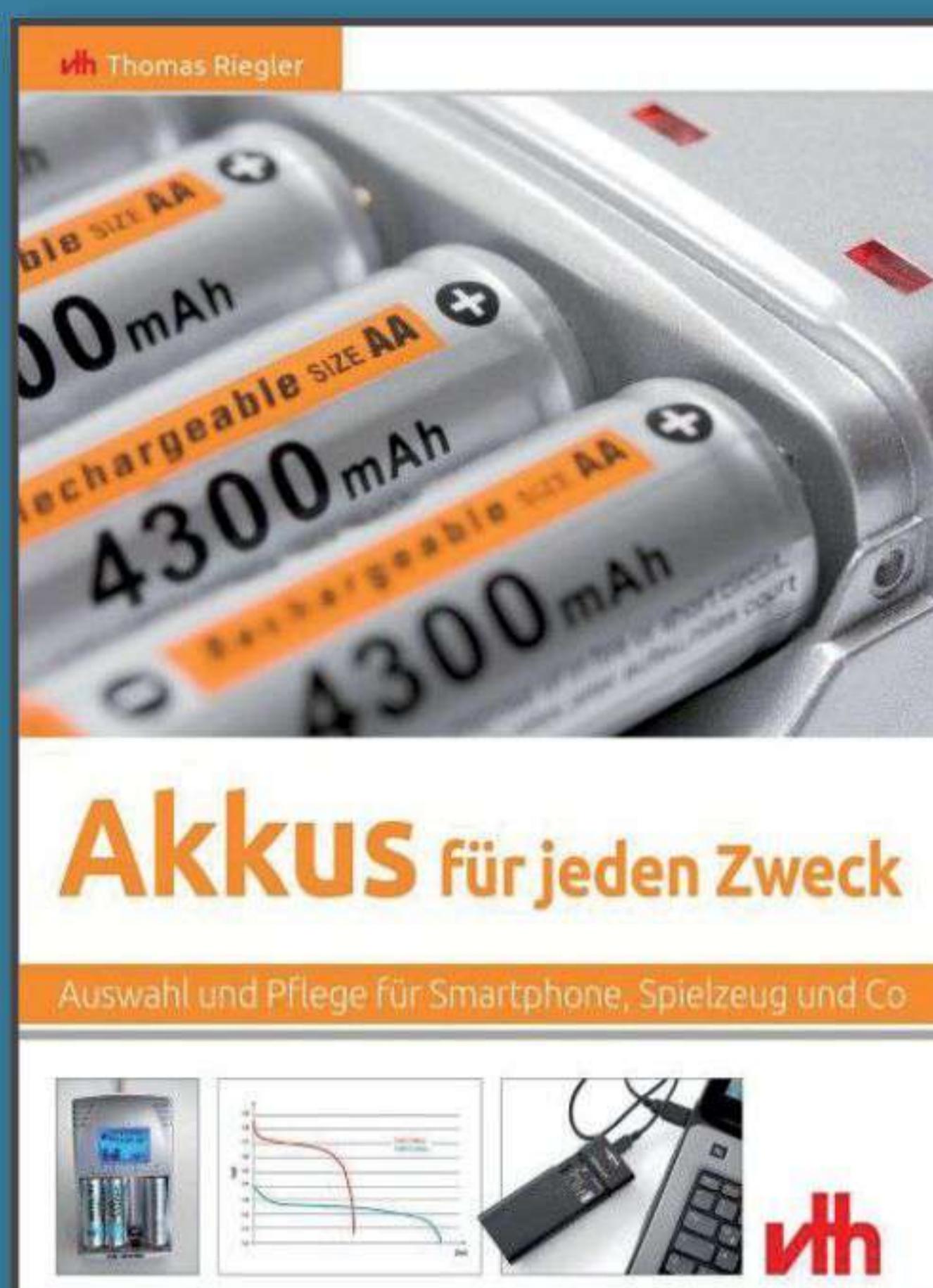
Für Windows 32-Bit, beispielsweise Windows 7 in der 32-Bit-Version, gibt es eine spezielle Variante dieses Programms unter der Bezeichnung *OnlineRechnerVerz32.exe*. Diese benötigt das ältere .NET-Framework in der Version 4. Letzteres ist meist schon auf dem Windows-PC installiert. Falls nicht, muss es installiert werden. Beide Versionen finden Sie auf der virtuellen DVD zum Download im Ordner \Artikel\OnlineRechner.

Michael Wöste, DL1DMW

Unsere Bibliothek



**HÖRZU Radio Guide -
30 Jahre Jubiläum**
Autor: Gerd Klawitter
ArtNr: 4130030
Preis: 26,90 €



Akkus für jeden Zweck

Auswahl und Pflege für Smartphone, Spielzeug und Co.



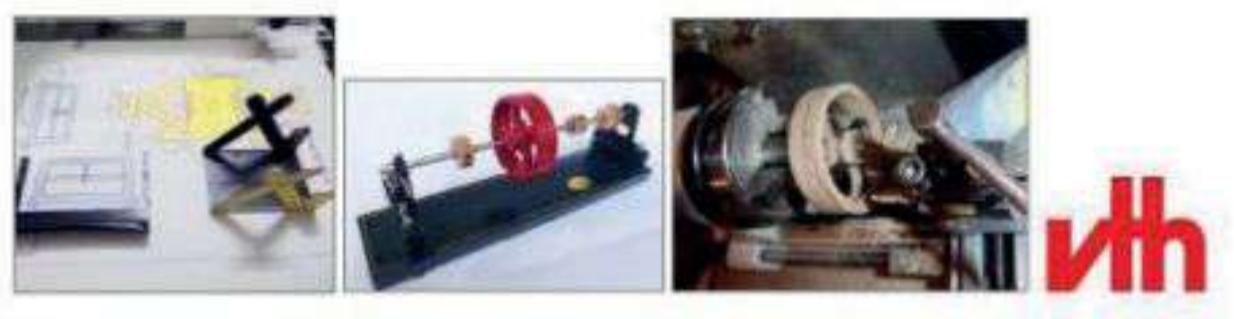
Akkus für jeden Zweck
Autor: Thomas Riegler
ArtNr: 3102283
Preis: 29,90 €



ANTRIEBSMODELLE

für Dampfmaschinen und Heißluftmotoren

Mit einfachen Mitteln selbst gebaut



**Antriebsmodelle für
Dampfmaschinen und
Heißluftmotoren**
Autor: Volker Koch
ArtNr: 3102295
Preis: 29,90 €



**50 Kniffe
für die Werkstatt**

Tipps und Tricks für den Modellbauer



**50 Kniffe für die
Werkstatt**
Autor: Kurt Becke
ArtNr: 3102289
Preis: 19,90 €

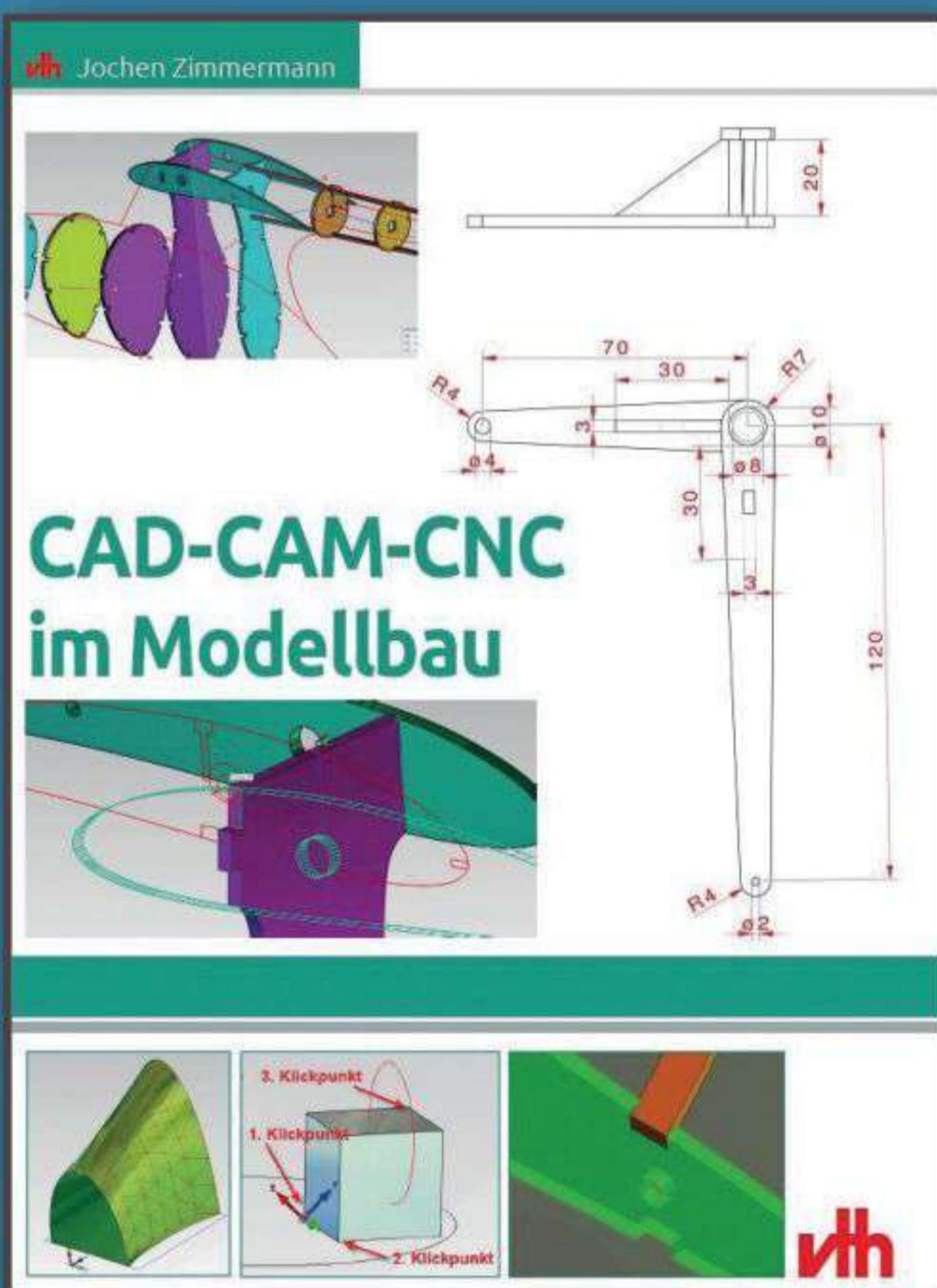


Elektrofeinwerkzeuge

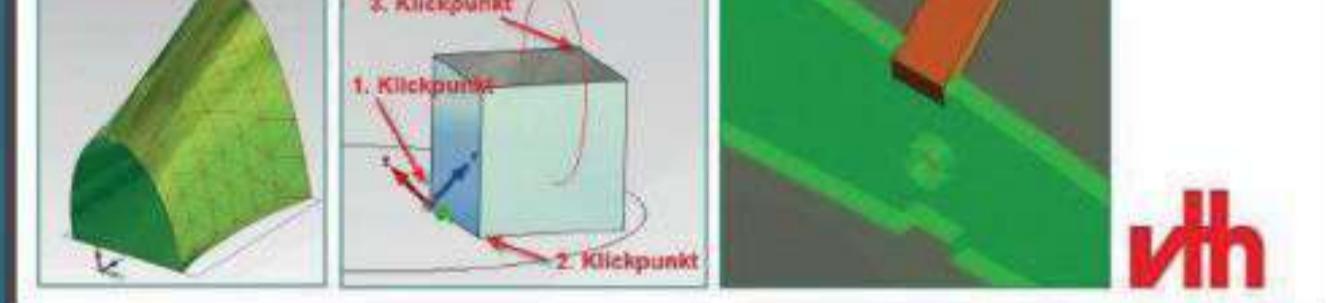
Geräte & Praxis



Elektrofeinwerkzeuge
Autor: Thomas Riegler
ArtNr: 3102263
Preis: 29,80 €



**CAD-CAM-CNC
im Modellbau**

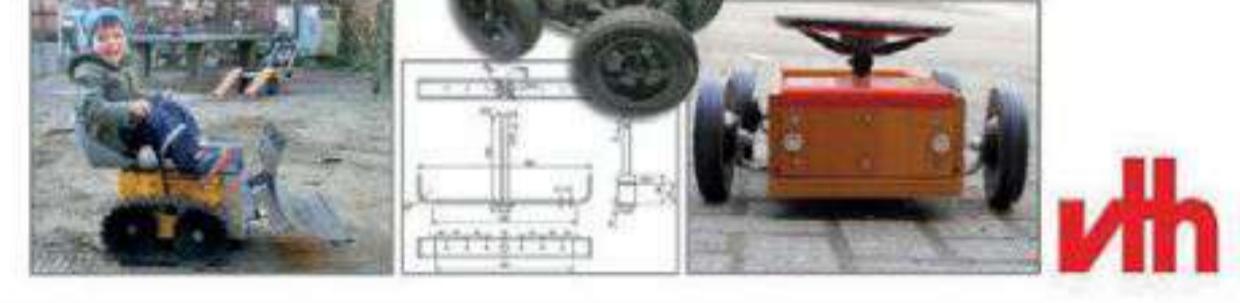


**CAD - CAM - CNC
im Modellbau**
Autor: Jochen Zimmermann
ArtNr: 3102270
Preis: 32,90 €

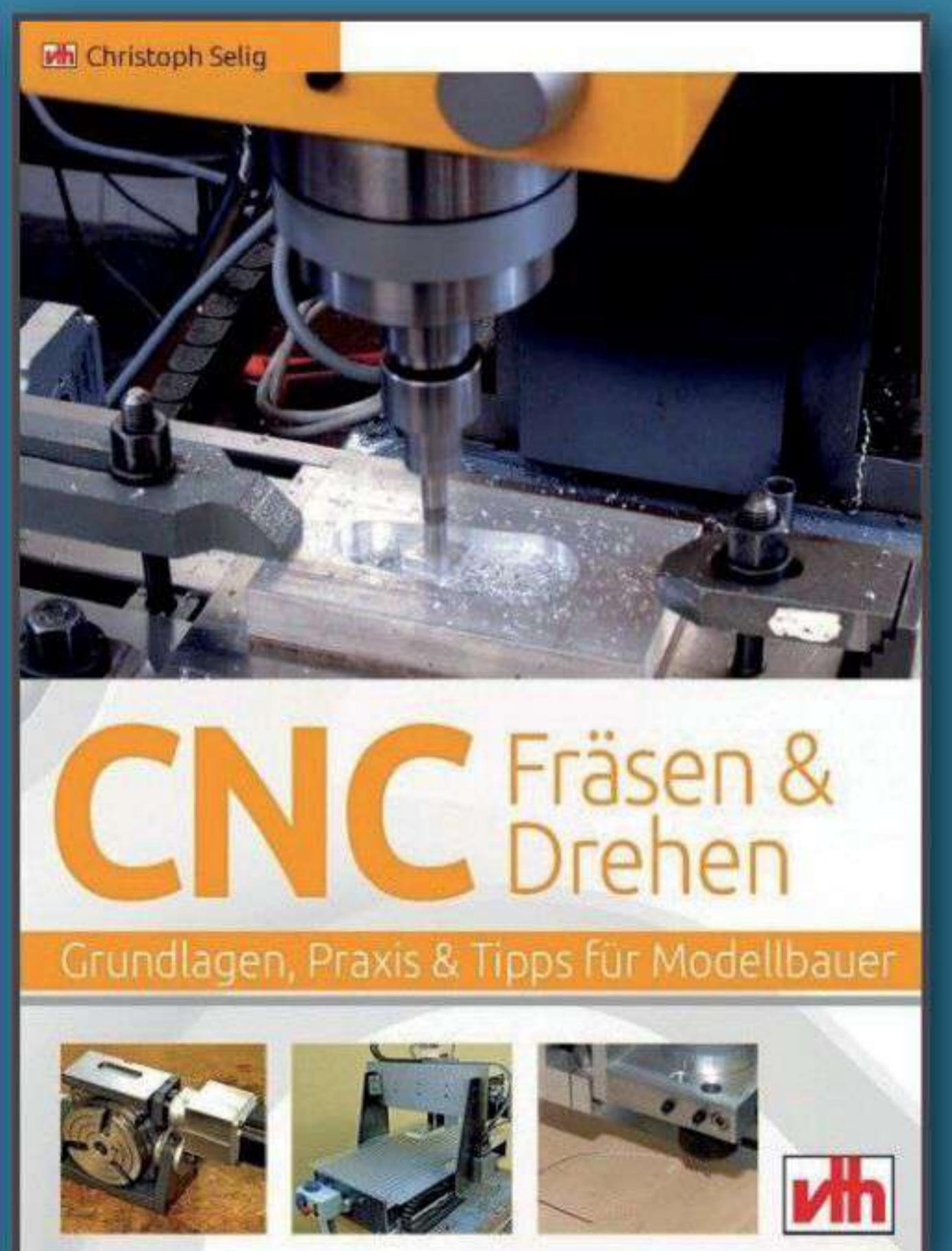


Tuning für Kinderautos

Eigenbau, Umbau und Motorisierung von Kinderfahrzeugen

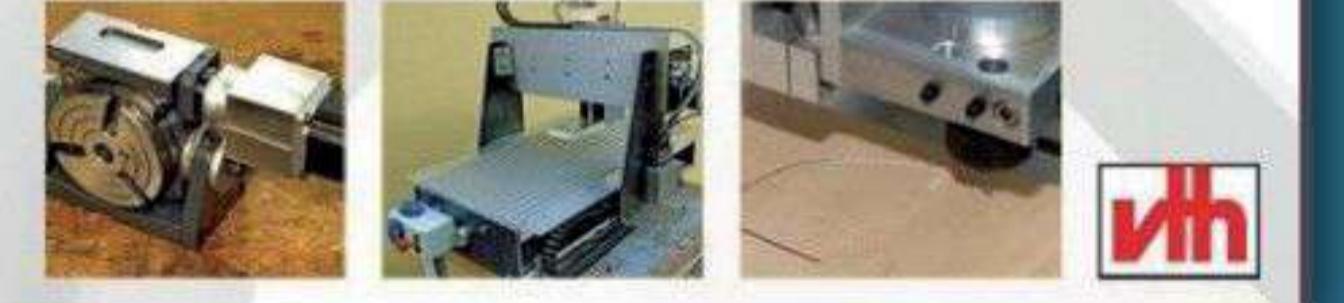


Tuning für Kinderautos
Autor: Tom Masselter
ArtNr: 3102293
Preis: 19,90 €



**CNC Fräsen &
Drehen**

Grundlagen, Praxis & Tipps für Modellbauer



CNC Fräsen & Drehen
Autor: Christoph Selig
ArtNr: 3102256
Preis: 31,90 €

Jetzt bestellen!

07221 - 5087-22
07221 - 5087-33
service@vth.de

www.vth.de/shop

[vth_modellbauwelt](https://www.instagram.com/vth_modellbauwelt)

[VTH neue Medien GmbH](https://www.youtube.com/VTH_neue_Medien_GmbH)

Bücher & Zeitschriften

PORTOFREI

(innerhalb Deutschland)

VTH & FMT
 VTH Verlag

vth

Vergriffen? Nicht bei uns!

PRINT ON DEMAND

Sie wünschen, wir drucken.

Mit Print on Demand produzieren wir Fachliteratur sofort nach Bestelleingang.



ArtNr: 4110173

Preis: 49,90 €



ArtNr: 4110115

Preis: 29,90 €

Dieser Button kennzeichnet unsere „Print on Demand - Produkte“

**PRINT
ON
DEMAND**



ArtNr: 4110036

Preis: 29,90 €



ArtNr: 4110171

Preis: 39,90 €



ArtNr: 4110124

Preis: 29,90 €

Jetzt bestellen!

07221 - 5087-22

07221 - 5087-33

service@vth.de

www.vth.de/shop

vth_modellbauwelt

VTH neue Medien GmbH

Bücher & Zeitschriften

PORTOFREI

(innerhalb Deutschland)

VTH & FMT
 VTH Verlag

vth